

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Doble Grado en Ingeniería Informática y Matemáticas

TRABAJO FIN DE GRADO

**Modelado e Implementación de Algoritmos de Movimiento para
Robots Holonómicos de Tres Ruedas**

José Ignacio de la Serna Cabello
Tutor: Fernando Jesús López Colino

Junio 2018

Modelado e Implementación de Algoritmos de Movimiento para Robots Holonómicos de Tres Ruedas

AUTOR: José Ignacio de la Serna Cabello

TUTOR: Fernando Jesús López Colino

**Dpto. Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2018**

Resumen

Este trabajo ha consistido en el desarrollo e implementación de un modelo de control para el movimiento de un robot omnidireccional de estructura rígida con tres ruedas. Ha tenido como objetivo que el robot autorregule su desplazamiento, y no haya que hacer nada más que indicarle el punto de destino.

Tras la introducción en los dos primeros capítulos, donde se presenta el tema y se exponen los principales conceptos relacionados con el modelado de algoritmos para el movimiento de robots móviles, en las secciones de diseño y de desarrollo se describe en profundidad el proyecto realizado.

Es en la sección de diseño donde se explican los modelos que subyacen bajo las funciones. Éstos son dos: el modelo cinemático inverso, que con la geometría del robot determina la velocidad de cada rueda, para moverse hasta la posición destino; y el modelo de los motores, que controla el funcionamiento de éstos y los regula según sea necesario.

En el desarrollo se analizan las cuestiones surgidas a lo largo de toda la elaboración del proyecto. Que son, la concepción de la máquina de estados para medir la velocidad de las ruedas, determinar la frecuencia más adecuada para computar la velocidad, cómo calcular la función de transferencia y la adaptación del paper.

Finalmente se describen las pruebas llevadas a cabo, junto con la discusión de los resultados, y la conclusión, donde se comentan las carencias del modelo utilizado. Los resultados obtenidos han obligado a la realización de cambios para un correcto funcionamiento, dichos cambios se aclaran en esa misma sección.

Palabras clave

Robot, omnidireccional, holonómico, encoder, regulador, modelo cinemático, función de transferencia, jacobiano, vector.

Abstract

This Bachelor Thesis consists in the development and implementation of a control model for the movement of an omnidirectional robot with a rigid structure with three wheels. Its aim is for the robot to self-regulate its own movement, and for it no longer to be necessary to do anything more than to indicate the point of destination.

After the introduction in the first two chapters, where the subject is presented and the main concepts related to the model algorithms for the movement of mobile robots are laid out, in the sections design and development the project carried out is described in depth.

It is in the design section where the models underlying the functions are explained. These are two: the inverse kinematic model, which with the geometry of the robot determines the speed of each wheel, so that to move to the destination position; and the model of the engines, which controls the behaviour of these and regulates them as necessary.

In the development, all the issues arising throughout the elaboration are analysed. These include, the conception of the state machine to measure the speed of the wheels, determine the most suitable frequency to compute the speed, how the transfer function is to be calculated, and the arrangement of the paper.

Finally, the tests carried out are described, along with the discussion of the results, and the conclusion, where the shortages of the model used are discussed. The obtained results have inevitably led to make changes for a correct behaviour, these changes are clarified in that same section.

Keywords

Robot, omnidirectional, holonomic, encoder, engine controller, kinematic model, transfer function, jacobian, vector.

Agradecimientos

A mis padres.

A mis amigos.

A mis profesores.

A todas aquellas personas que me he encontrado.

Deo Gratias.

INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos.....	1
1.3	Organización de la memoria.....	1
2	Estado del arte	3
3	Diseño.....	5
3.1	Introducción.....	5
3.2	Modelo.....	7
3.2.1	Modelo cinemático	7
3.3	Motores.....	8
3.3.1	Controlador.....	8
3.3.1	Encoders	9
3.4	Programa principal	10
4	Desarrollo	11
4.1	Modelo motores.....	11
4.1.1	Motores y codificadores	11
4.1.2	Frecuencia de muestreo	12
4.1.3	Función de transferencia.....	13
4.1.4	Reguladores	15
4.2	Modelo robot	15
4.2.1	Adaptación del paper	15
5	Integración, pruebas y resultados	19
1.1.	Integración.....	19
1.2.	Pruebas y resultados	19
6	Conclusiones y trabajo futuro.....	24
	Referencias	25
	Glosario	27
	Anexos.....	I
A	Pruebas con varias frecuencias	I
B	Medidas ruedas	VI
B.1	Tablas	VI
(1)	Rueda 1	VI
(2)	Rueda 2	VII
(3)	Rueda 3	VIII
B.2	Gráficas.....	IX
(1)	Rueda 1	IX
(2)	Rueda 2	XIII
(3)	Rueda 3	XVI
C	Ajuste de las gráficas	XX
C.1	Rueda 1	XX
C.2	Rueda 2	XXI
C.3	Rueda 3	XXII

INDICE DE FIGURAS

FIGURE 1: VISTA EN PLANTA DEL ROBOT	5
FIGURE 2: SISTEMA DE CONTROL DE LAZO CERRADO	9
FIGURE 3: COMPARACIÓN DE LAS VELOCIDADES ANTES Y DESPUÉS DE NORMALIZAR	17
FIGURE 4: SISTEMA DE COORDENADAS DEL PAPER	18
FIGURE 5: SISTEMA DE COORDENADAS DESEADO	18
FIGURE 6: DISPOSICIÓN DEL ROBOT EN LAS PRUEBAS	19

INDICE DE ILUSTRACIONES

ILUSTRACIÓN 2-I: IMAGEN DE UNA RUEDA OMNIDIRECCIONALORIENTADA 90°	4
ILUSTRACIÓN 4-I: ENCODER OUTPUT (FTE. WWW.ELECTRONICS-TUTORIALS.WS)	11
ILUSTRACIÓN 5-I: VECTORES QUE INDICAN LAS POSICIONES DESEADAS	20
ILUSTRACIÓN 5-II: VECTORES DE POSICIÓN OBTENIDOS EN LAS PRIMERAS PRUEBAS	21
ILUSTRACIÓN 5-III: VECTORES DE POSICIÓN OBTENIDOS TRAS LA MODIFICACIÓN	22
ILUSTRACIÓN 5-IV : ERROR DE PRECISIÓN DEL MOVIMIENTO	23

INDICE DE IMÁGENES

IMAGEN 1: FOTO DEL ROBOT UTILIZADO, HCTRON HOLOBOT 1.0	6
--	---

INDICE DE GRÁFICAS

GRÁFICA 1: AJUSTE DE LA GRÁFICA DE LA FUNCIÓN DE LA RUEDA 1	14
GRÁFICA 2: ERROR DEL AJUSTE DE LA FUNCIÓN DE LA RUEDA 1	14
GRÁFICA 3: 20 HZ	I
GRÁFICA 4: 50 HZ	I
GRÁFICA 5: 100 HZ	II
GRÁFICA 6: 150 HZ	II

GRÁFICA 7: 200 0HZ	III
GRÁFICA 8: 250 HZ	III
GRÁFICA 9: 300 HZ	IV
GRÁFICA 10: 350 HZ	IV
GRÁFICA 11: 400 HZ	V
GRÁFICA 12: 500 HZ	V
GRÁFICA 13: RUEDA 1, VEL. 50, 3ª	IX
GRÁFICA 14: RUEDA 1, VEL. 50, 1ª	IX
GRÁFICA 15: RUEDA 1, VEL. 50, 2ª	IX
GRÁFICA 16:RUEDA 1, VEL. 75, 2ª	X
GRÁFICA 17: RUEDA 1, VEL. 50, 4ª	X
GRÁFICA 18: RUEDA 1, VEL. 75, 1ª	X
GRÁFICA 19: RUEDA 1, VEL. 75, 13	XI
GRÁFICA 20: RUEDA 1, VEL. 90, 1ª	XI
GRÁFICA 21: RUEDA 1, VEL. 75, 4ª	XI
GRÁFICA 22: RUEDA 1, VEL. 90, 4ª	XII
GRÁFICA 23: RUEDA 1, VEL. 90, 2ª	XII
GRÁFICA 24: RUEDA 1, VEL. 90, 3ª	XII
GRÁFICA 25: RUEDA 2, VEL. 50, 1ª	XIII
GRÁFICA 26: RUEDA 2, VEL. 50, 3ª	XIII
GRÁFICA 27: RUEDA 2, VEL. 50, 2ª	XIII
GRÁFICA 28: RUEDA 2, VEL. 75, 1ª	XIV
GRÁFICA 29: RUEDA 2, VEL. 50, 4ª	XIV
GRÁFICA 30: RUEDA 2, VEL. 75, 2ª	XIV
GRÁFICA 31: RUEDA 2, VEL. 75, 3ª	XIV
GRÁFICA 32 : RUEDA 2, VEL. 75, 4ª	XIV

GRÁFICA 33: RUEDA 2, VEL. 90, 1ª	XV
GRÁFICA 34: RUEDA 2, VEL. 90, 2ª	XV
GRÁFICA 35: RUEDA 2, VEL. 90, 4ª	XV
GRÁFICA 36: RUEDA 2, VEL. 90, 3ª	XV
GRÁFICA 37: RUEDA 3, VEL. 50, 1ª	XVI
GRÁFICA 38: RUEDA 3, VEL. 50, 3ª	XVI
GRÁFICA 39: RUEDA 3, VEL. 50, 2ª	XVI
GRÁFICA 40: RUEDA 3, VEL. 75, 2ª	XVII
GRÁFICA 41: RUEDA 3, VEL. 50, 4ª	XVII
GRÁFICA 42: RUEDA 3, VEL. 75, 1ª	XVII
GRÁFICA 43: RUEDA 3, VEL. 75, 3ª	XVIII
GRÁFICA 44: RUEDA 3, VEL. 75, 2ª	XVIII
GRÁFICA 45: RUEDA 3, VEL. 90, 1ª	XVIII
GRÁFICA 46: RUEDA 3, VEL. 75, 2ª	XIX
GRÁFICA 47: RUEDA 3, VEL. 75, 3ª	XIX
GRÁFICA 48: RUEDA 3, VEL. 75, 4ª	XIX
GRÁFICA 49: AJUSTE DE LA GRÁFICA DE LA FUNCIÓN DE LA RUEDA 1	XX
GRÁFICA 50: ERROR DEL AJUSTE DE LA FUNCIÓN DE LA RUEDA 1	XX
GRÁFICA 51: AJUSTE DE LA GRÁFICA DE LA FUNCIÓN DE LA RUEDA 2	XXI
GRÁFICA 52: ERROR DEL AJUSTE DE LA FUNCIÓN DE LA RUEDA 2	XXI
GRÁFICA 53: AJUSTE DE LA GRÁFICA DE LA FUNCIÓN DE LA RUEDA 3	XXII
GRÁFICA 54: ERROR DEL AJUSTE DE LA FUNCIÓN DE LA RUEDA 3	XXII

1 Introducción

1.1 Motivación

Cada vez está más presente la tecnología en todos los ámbitos de nuestra existencia, desde los lugares más íntimos y personales, hasta los espacios casi desconocidos para nosotros como otros planetas. Esta irrupción de la tecnología facilita el avance y desarrollo, rompiendo barreras y abriendo nuevos caminos, buscando llevar los límites conocidos un paso más allá y explorando nuevas formas de innovación.

En este contexto la robótica móvil ha comenzado a ser un campo cada vez más investigado, debido a sus importantes implicaciones para la navegación autónoma, o simplemente para explorar leyes de control en el movimiento autónomo, que se pueden aplicar en campos como la salud, seguridad, espacio... Sus aplicaciones están relacionadas con tareas que implican un riesgo, requieren un alto grado de precisión o facilitan el trabajo a los humanos y aumentan la eficiencia.

El principal objetivo a resolver de un robot móvil es el de generar trayectorias factibles y guiar su movimiento. Afrontar dicho propósito exige distintos niveles de procesamiento de la información, que tienen distintos grados de complejidad. La navegación más básica utiliza simplemente un modelo cinemático, de baja complejidad y que no tiene en cuenta los elementos de su entorno.

1.2 Objetivos

El objetivo de este TFG es la generación e implementación de un sistema cinemático que controle el movimiento de un robot holonómico de tres ruedas. La realización de este sistema tiene como finalidad que el robot se mueva a los puntos dados en coordenadas. Para que el modelo cinemático se comporte correctamente será necesario también modelar los motores.

Se ha trabajado con un robot de tres ruedas omnidireccionales fijas, colocadas en ángulos de 60 grados. Debido a la disposición de las ruedas y las características de éstas, estamos ante un robot holonómico en dos dimensiones, lo que significa que el robot tiene movilidad en todas las direcciones, y por tanto las trayectorias calculadas serán en línea recta.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Introducción:** Es la sección presente, en ella se describen la motivación para la realización de este TFG, los objetivos del mismo y se presenta la estructura que va a seguir la memoria.

- **Estado del arte:** En esta sección se pretende ubicar al lector en el ámbito en que desarrolla el trabajo. Dando explicación de los tipos de enfoque que se pueden dar en el ámbito de la robótica móvil, y las formas de abordar el desarrollo de los módulos de control de un robot móvil.
- **Diseño:** Aquí se explica la organización e implementación de los módulos que conforman el sistema de control, desarrollando cada función y comentando su funcionamiento.
- **Desarrollo:** Con este capítulo se pretende ilustrar el trabajo elaborado y el porqué de ciertas decisiones de implementación tomadas. Así como, las dificultades que se han encontrado y las principales cuestiones abordadas a lo largo del desarrollo.
- **Integración y pruebas:** Describe el proceso de unificación del código y la posterior batería de pruebas.
- **Conclusiones y trabajo futuro:** En la conclusión se resaltan los aspectos más importantes del trabajo, y menciona las posibles líneas de continuación para mejorar y añadir nuevas funcionalidades.

2 Estado del arte

Este TFG se desarrolla en el marco de la robótica móvil, más concretamente en el desarrollo de algoritmos de control de movimiento. La misión de estos algoritmos de control es comparar el estado actual con el estado deseado, y realizar las acciones necesarias para su consecución.

Los primeros vehículos guiados automáticamente aparecieron para proporcionar un sistema de transporte de materiales dentro de las fábricas entre las distintas cadenas de producción, utilizando primero carriles y después sustituidos por una red de guiado instalada en el suelo. Estos tenían una capacidad perceptual y de decisión mínima, y ante cualquier evento inesperado el vehículo se bloqueaba quedando incapaz de reanudar la tarea. Por ese motivo surgieron los robots móviles, capaces de moverse en entornos dinámicos.

Un aspecto importante para dotar de autonomía a los robots son los modelos matemáticos que controlan el comportamiento del robot, éstos varían en función de la finalidad que tienen, y su complejidad dependerá de la dificultad del propósito. Se pueden distinguir dos medios de aproximación al problema: los basados en un modelo cinemático, que consideran puramente la geometría del movimiento a lo largo del tiempo (posición, orientación, velocidad), y los basados en un modelo dinámico, que además tienen en cuenta las fuerzas del movimiento (masa, inercia, fricción...).

La cinemática del robot consiste en estudiar la variación de su posicionamiento con respecto a un sistema de referencia y su relación con los valores de sus coordenadas articulares (por ejemplo, la velocidad de las ruedas).

- Cinemática directa: modelos de ecuaciones cinemáticas que permiten el cálculo de los vectores de posición de los elementos del robot en función del tiempo, respecto a un sistema de referencia, a partir de las velocidades de movimiento de los actuadores.
- Cinemática inversa: son los modelos con los que se obtienen las velocidades de movimiento de los actuadores, teniendo como medidas la posición final de los elementos del robot.

Asumiendo que no existe deslizamiento de las ruedas y que el terreno es totalmente plano, el uso del modelo cinemático para el desarrollo de algoritmos de control es convenientemente útil, ya que resultan relativamente simples y fáciles de implementar.

Las características de un robot móvil varían en función de la tarea para la que haya sido específicamente diseñado. Éstas pueden ser tanto estructurales: forma del robot, número y tipo de rueda; como sensoriales: acelerómetro, sensores ópticos...

Hay robots que tienen que tener en cuenta la tracción por la necesidad de controlarlos en terrenos poco compactos, como arena, barro, nieve... o en terreno irregular donde no llega a estar en contacto con el suelo, como terrenos rocosos. El uso de modelos cinemáticos puede provocar errores de posicionamiento del robot, especialmente en aquellos que sólo utilizan codificadores ópticos incrementales instalados en las ruedas para estimar el desplazamiento lineal.

A parte de la clasificación algorítmica, en los robots existen otras clasificaciones, la que a nosotros nos interesa tiene que ver con los grados de libertad de movimiento. En este sentido se los puede dividir en dos clases:

- Robots holonómicos, poseen los mismos grados de libertad que de control. En la superficie poseen total movilidad, es decir, en cualquier instante son capaces de moverse en cualquier dirección, independientemente de su orientación.
- Robots no holonómicos: el vehículo posee más grados de libertad que de control, lo que significa que en el plano no se puede mover en todas direcciones.

Los sistemas holonómicos con capacidad de moverse en cualquier dirección los hace muy atractivos en ambientes dinámicos donde una gran capacidad de movilidad y respuesta es deseable.

Este trabajo está centrado en un robot omnidireccional u holonómico, de estructura rígida que posee tres ruedas fijas, es decir no tiene ningún elemento articulador. Para que el robot tenga plena movilidad es imprescindible que las ruedas permitan movimiento en todas las direcciones, y al estar fijas sólo es posible si son omnidireccionales. Las ruedas son ruedas fijas con rodillos orientados 90° .



Ilustración 2-I: Imagen de una rueda omnidireccional orientada 90°
(fte. www.active-robots.com)

3 Diseño

3.1 Introducción

En este capítulo se describen los tres módulos en los que se ha dividido el trabajo desarrollado sobre los algoritmos de control del robot.

Estos tres módulos se han llevado a cabo de forma secuencial y coinciden con las fases de desarrollo del proyecto. Sin embargo, los presentes módulos no están ordenados de acuerdo al criterio del tiempo, como sí lo está el punto 4 Desarrollo, si no que van desde una visión más amplia y completa de la cuestión hacia aspectos más específicos.

Para abordar el tema hace falta aclarar los elementos que intervienen en el dinamismo del robot.

Primero, si queremos ordenar al robot que se desplace, antes es necesario que éste se pueda situar. Para eso elegimos el siguiente sistema cartesiano con origen de coordenadas en el centro del robot, que sirva de referencia:

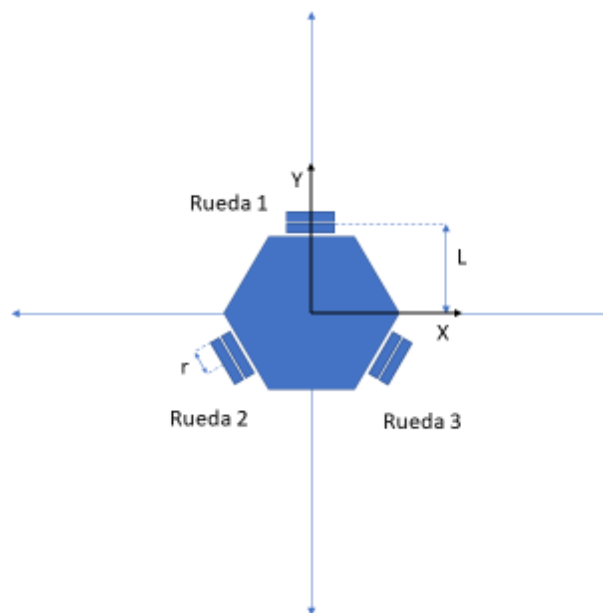


Figure 1: Vista en planta del robot

El robot tiene 3 ruedas fijas de tipo sueca (con rodillos orientados 90°), dispuestas paralelas su lado opuesto, en tres lados intercalados del hexágono formado por la placa base del robot. El radio de cada rueda (r) mide 0,028 metros y la mitad de la altura (L) 0,082 metros.

El giro de las ruedas se medirá en pasos, unidad que corresponderá al cambio de fase de las señales que nos llegan de los encoders. En el punto de Desarrollo se explicará más en profundidad, y se dirá la conversión que tiene con los radianes.

A continuación se muestra una foto desde el plano cenital, del robot en el suelo del laboratorio. En ella se pueden ver las ruedas omnidireccionales, que están dispuestas formando un triángulo equilátero.

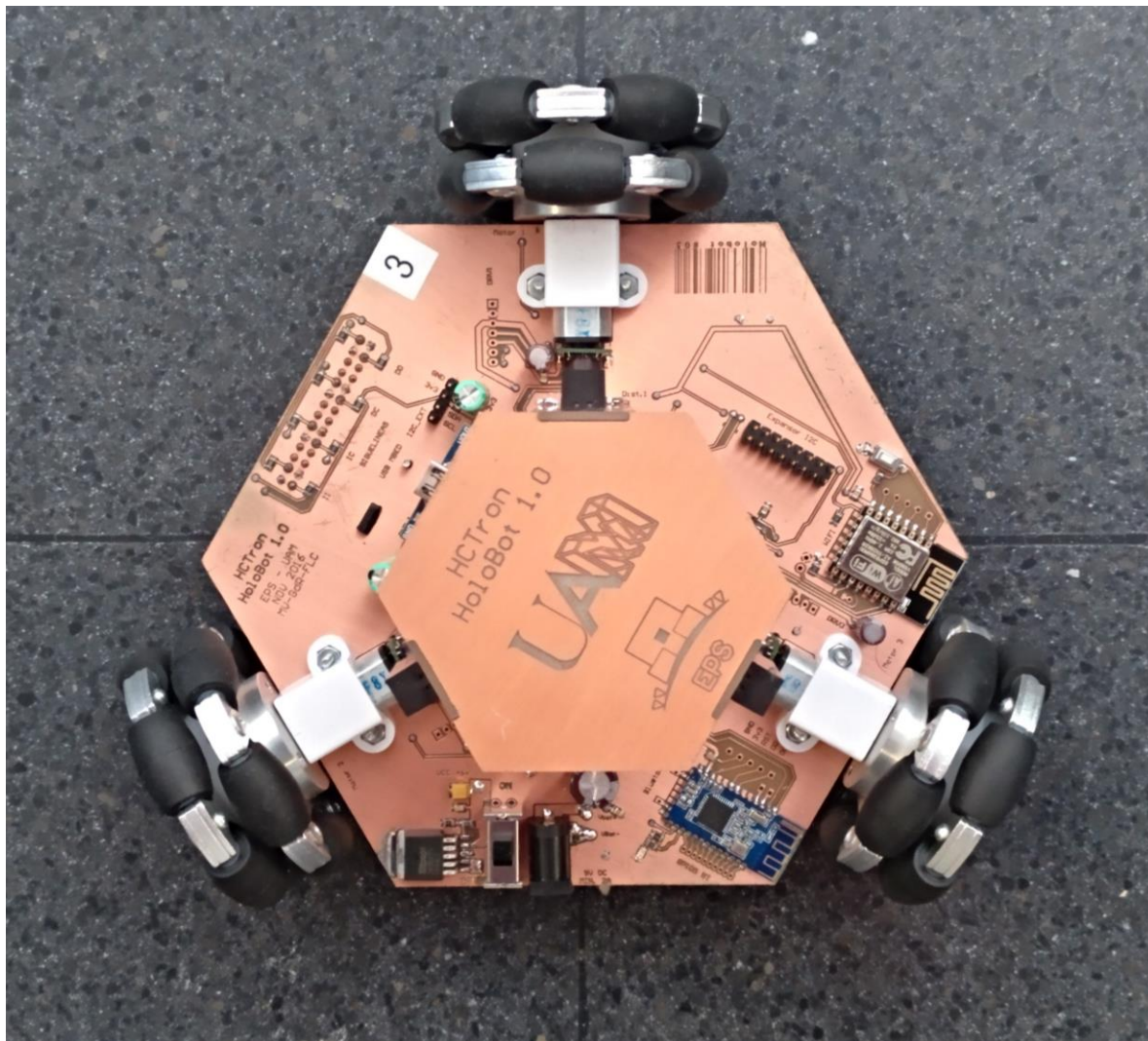


Imagen 1: Foto del robot utilizado, HCTron Holobot 1.0

3.2 Modelo

3.2.1 Modelo cinemático

Este es el módulo que computa la velocidad a la que se debe mover cada rueda. Está basado en el modelo del paper *Modelado cinemático y dinámico de un robot móvil omnidireccional* [1]. En dicho modelo se hacen todos los cálculos necesarios para la obtención de las velocidades de cada una de las 3 ruedas del robot dadas las indicaciones de posicionamiento deseado. Concretamente se ha utilizado una variación de la fórmula (33), que es la matriz inversa del jacobiano del vehículo.

El jacobiano o matriz jacobiana del robot se define como el modelo que relaciona las variaciones entre las posiciones y las orientaciones de un sistema de referencia con las variables de control.

A continuación el jacobiano inverso:

$$\begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} = 1/r \cdot \begin{pmatrix} \cos \Theta_1 & \sin \Theta_1 & -L \\ -\sin \Theta_2 & \cos \Theta_2 & -L \\ -\sin \Theta_3 & -\cos \Theta_3 & -L \end{pmatrix} \cdot \begin{pmatrix} V_{Cx} \\ V_{Cy} \\ \omega_C \end{pmatrix}$$

Donde V_{Cx} y V_{Cy} son las proyecciones en los ejes de la velocidad lineal del robot, que equivale al vector de posición deseado respecto del centro del robot. ω_C es la velocidad angular de rotación sobre sí mismo. ω_{1x} , ω_{2x} y ω_{3x} son las velocidades angulares de cada rueda. r es el radio de las ruedas, y L es la distancia de las ruedas al centro del robot (C).

El vector de velocidad, dado que no varía, podemos decir que es equivalente al vector de posición. Y entonces podemos sustituir en la formula donde corresponde la velocidad en el eje x por la coordenada x de la posición a la que deseamos que se desplace el robot, y la velocidad en el eje y por la coordenada y de la posición deseada.

Así, el modelo inverso permite determinar qué valores hay que aplicar a las variables de control (velocidad angular de las ruedas, en nuestro caso aplicada al motor como consignas de velocidad), para que la posición y orientación respecto del sistema de referencia sean las deseadas.

Este modelo está implementado en las funciones `vel_angular1`, `vel_angular2` y `vel_angular3`, que calculan la velocidad angular de cada rueda dada la posición final y la velocidad angular de rotación del robot sobre su centro.

$$\begin{aligned} \omega_1 &= (x + -0.082 \cdot z) \cdot \frac{1}{0.028} \\ \omega_2 &= \left(-\frac{1}{2}x + \frac{\sqrt{3}}{2}y - 0.082 \cdot z \right) \cdot \frac{1}{0.028} \\ \omega_3 &= \left(-\frac{1}{2}x - \frac{\sqrt{3}}{2}y - 0.082 \cdot z \right) \cdot \frac{1}{0.028} \end{aligned}$$

Pero esto es un modelo teórico y no tiene en cuenta las limitaciones físicas del robot, es por esto que son necesarias correcciones adicionales que garanticen el justo funcionamiento del robot.

En la función normalizar se comprueba si alguna de las velocidades angulares supera la velocidad máxima de rotación del motor, y si es así, normaliza los valores utilizando el máximo de esos valores. La velocidad máxima calculada es 10 radianes por segundo.

A continuación se muestra el cálculo de la velocidad máxima en radianes:

- Número de pasos por revolución: 1800 pasos.
- Velocidad máxima en pasos: aproximadamente 2900 pasos por segundo.
- Hallamos la frecuencia máxima:

$$frecuencia_{max} = Velocidad_{max} / Pasos \text{ por revolución} = 2900 / 1800 = 1,6111$$

$$(pasos/segundo) / (pasos/revolución) = (revoluciones/segundo)$$

- Velocidad máxima en radianes:

$$\omega = 2\pi f = 10,123 \text{ rad/s}$$

La velocidad calculada con el modelo se hace en radianes por segundo, pero al motor hay que mandársela en pasos, no en radianes. Con la función `vel_pasos`, se transforma la velocidad angular a pasos por 0,005 segundos, unidad de medida que utiliza el regulador de los motores.

Y la función principal de este módulo se llama *modelo*. Ésta se encarga de llamar a las funciones anteriores para transformar las coordenadas dictadas al robot en consignas para los motores. Primero calcula las velocidades angulares dadas las coordenadas, en el segundo paso las normaliza, y por último les cambia de unidad de medida a pasos por 0,005 segundos.

3.3 Motores

Este módulo se divide en dos secciones: el controlador y los encoders. El primero se encarga de controlar que el sistema responde como debe hacerlo y tomar las acciones adecuadas, y el segundo sirve de ayuda al controlador siendo el encargado de procesar las señales de los encoders.

3.3.1 Controlador

El regulador es un sistema de control que regula el comportamiento del motor. Se ha implementado un sistema de lazo cerrado, o sea, que el regulador hace correcciones en la señal enviada al motor en función de la salida obtenida, para que se cumpla la consigna deseada.

Regulación lazo cerrado



Figure 2: Sistema de control de lazo cerrado

En la imagen se aprecia la simplicidad del sistema que conforma el regulador. El regulador toma la salida del motor y la consigna de velocidad, y hace los cálculos adecuados para enviar la nueva consigna al motor y que se ajuste más la salida a la esperada.

El controlador se articula con 3 funciones:

- *start_ruedas*: Es la función principal. Inicializa las variables de las funciones del regulador, y de las ruedas. También se encarga de controlar el tiempo de movimiento del robot. Esto la convierte en la función que conecta y pone en marcha todo el mecanismo.
- *timeInterrupt*: Función que se inicializa desde el módulo de los encoders. Esta función se ejecuta cada 5 milisegundos y es la que se encarga de llamar a la función regulador de cada rueda, que se describe a continuación. De esta manera, la entrada al motor es ajustada cada 5 milisegundos.
- *regulador1*: Hace los cálculos del regulador. Con la consigna (velocidad a la que se desea que se mueva el motor) y con la salida del motor (velocidad efectiva obtenida con los encoders) calcula la nueva velocidad que asignar al motor y se la manda. Esta función es llamada cada 5 milisegundos por la función *timeInterrupt*.

3.3.1 Encoders

El módulo de los encoders se ocupa computar el giro de las ruedas. Esto se hace sirviéndose de las señales que envían los encoders de las ruedas, que mediante una máquina de estados son procesadas de tal manera que se puede saber, dependiendo del estado anterior, si la rueda ha avanzado o retrocedido, y por tanto determinar su evolución, que comprobada cada cierto tiempo da lugar a su velocidad.

La función principal es *encoders*, que lo único que hace es establecer las interrupciones necesarias, y que son de dos tipos. Una es la interrupción asociada al tiempo de comprobación del avance: *timeInterrupt*, del módulo del controlador, y que salta cada 0,005 segundos. El segundo tipo de interrupción está asociado al flanco de subida o de bajada de las señales de los encoders. Interrupciones que saltan cuando se produce el flanco de

subida o de bajada, y que llaman a su correspondiente función de la máquina de estados, detalladas a continuación.

Las 4 funciones de cada rueda: RiseA1, FallA1, RiseB1 y FallB1 (en el caso de la rueda 1), conforman la máquina de estados de dicha rueda, y contabilizan los pasos. La máquina de estados es explicada en el primer apartado del siguiente punto, Desarrollo. De esta manera cada vez que salta la interrupción del tiempo (*timeInterrupt*), el controlador registra los pasos de cada rueda y pone sus contadores de nuevo a cero, adquiriendo la velocidad en pasos por 5 milisegundos.

Tras esta descripción se puede ver entonces, que este módulo una vez haya sido inicializado ya está preparado para cumplir su cometido: una vez que comienzan a girar las ruedas, ya se puede obtener su velocidad.

3.4 Programa principal

En este programa se introducen las coordenadas a las que tiene que ir el robot, y se llama a las funciones principales de los otros módulos, que son: *encoders*, *modelo* y *start_ruedas*.

El orden de llamadas es el siguiente: primero se llama a la función *encoders* para tener activa la lectura de la velocidad de giro de las ruedas; luego a *modelo*, para determinar la velocidad a la que debe girar cada una de las ruedas; y por último a *start_ruedas* que pone en marcha el regulador, el cual acciona las ruedas asignándoles la velocidad calculada con el modelo cinemático.

4 Desarrollo

4.1 Modelo motores

4.1.1 Motores y codificadores

Una vez entendida la API, la primera tarea ha consistido en la creación de un programa que llevase cuentas de la cantidad de giro de una rueda. Se ha procedido a medir el giro de la rueda con sus respectivos codificadores en cuadratura que utilizan un disco magnético y sensores de efecto Hall. El codificador genera dos señales digitales cuadradas A y B dependientes del campo magnético, con estas dos señales se puede medir la velocidad angular de una rueda.

La técnica para hacerlo es mediante la interfaz InterruptIn, que activa un evento cuando cambia la entrada de un pin digital. Así, definiendo este evento como la ejecución de una función, podemos registrar los cambios de las señales A y B que nos llegan a través de los pines de los codificadores.

Pero la velocidad no se puede medir simplemente midiendo el cambio de la señal respecto del tiempo, pues esto no garantiza saber si se está avanzando o retrocediendo, dando lugar a error. Es necesario realizar una máquina de estados que identifique si se ha realizado una transición hacia delante o hacia detrás. Cada una de las posibles combinaciones de salidas de A y B es un estado, 00, 01, 10 y 11, y en función del estado anterior podemos saber el sentido de avance.

En la siguiente imagen se muestra las secuencias de estados según el sentido de giro.

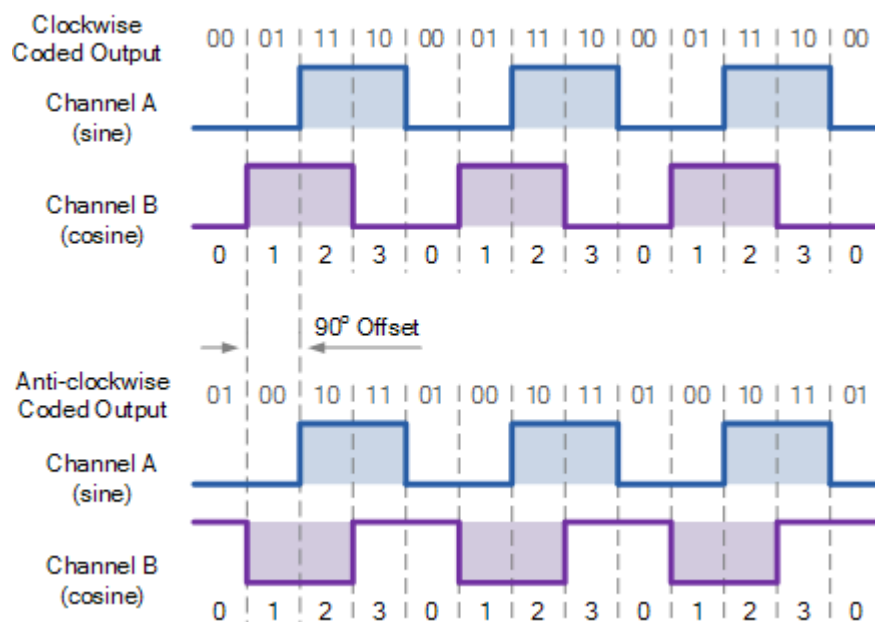


Ilustración 4-I: Encoder output (fte. www.electronics-tutorials.ws)

Como se puede ver, desde el 00 si el siguiente estado es 01 significa que se mueve en sentido horario, si es 10, antihorario. Esto es posible gracias al desfase de 90 grados que existe entre las dos señales.

Una vez resueltas las transiciones entre los estados, queda establecer el cálculo de la velocidad, con varias posibles interpretaciones. Midiendo la velocidad a través de la señal, podemos definir un paso o pulso como un periodo completo de oscilación, como la mitad del ciclo, o un cuarto (90°) que equivale al paso de un estado a otro.

Se han implementado versiones con todas las formas posibles de contar pasos y ha sido en el modelado de los motores donde se ha fijado que la opción más conveniente es aquella en la que un cambio de estado constituye un pulso. Siendo, por tanto, la velocidad en pulsos por segundo, equivalente a transiciones de estados por segundo.

Dado que el elemento InterruptIn permite la ejecución de una función cuando hay un cambio en la señal, se ha implementado un modelo en el que los estados quedan definidos por las transiciones, pasando así a aplicar un modelo más eficiente.

La descripción es la siguiente: si en la interrupción del flanco de subida de la señal A, la señal B es 1, entonces se cuenta un paso positivo, y si es 0, negativo; si en la interrupción del flanco de subida de la señal B, la señal A es 0, entonces es un paso positivo, y si es 1, negativo. Y de igual modo quedan designados con los flancos de bajada de ambas señales.

Esto supone una ventaja de tiempo y recursos, pues ya no es necesario mantener guardado el estado anterior, y simplemente es necesario hacer la comprobación del nivel en el que se encuentra la otra señal.

4.1.2 Frecuencia de muestreo

Para proceder al modelado de los motores es antes necesario establecer la frecuencia de muestreo para obtener la velocidad del motor.

El objetivo del modelado de los motores es encontrar una representación adecuada de su funcionamiento, para ello, se realizan pruebas y se toman medidas. Estas pruebas consisten en enviar al motor la consigna de velocidad a la que rotar y observar la evolución de su velocidad desde cero.

Pero a la hora de obtener el modelo, no da igual la frecuencia de toma de estas observaciones. El cambio de velocidad desde cero hasta la velocidad asignada sucede en un tiempo muy pequeño, así que cuanto mayor sea la velocidad de muestreo, mayor debería ser la precisión con la que captar el cambio. Sin embargo, la precisión no es infinita, existe un límite debido a que la velocidad se mide en pasos (unidad más pequeña de rotación de la rueda) por segundo, dando lugar a una frecuencia de muestreo óptima. Así, cuanto menor sea el tiempo entre una medida y la siguiente, menor número de pasos dará, y por tanto, que un paso sea contabilizado en un intervalo o en el siguiente, supondrá un porcentaje de error mayor.

Por ejemplo, con una velocidad de 1900 pasos por segundo, no es lo mismo medir en un intervalo de 2 milisegundos que en uno de 5 milisegundos, pues en el primero se contabilizan 3 o 4 pasos, y en el segundo 9 o 10 (no se puede medir ni 3.8 pasos ni 9.5 pasos). Y esto supone una variación de más del 25% frente a una que no llega al 11%.

Aquí se ve, que cuanto más pequeña sea la unidad *paso* como medida, mayor precisión tendremos. Es por este motivo, que la unidad de referencia de un paso se ha establecido como el cambio de estado de la máquina descrita en la figura anterior.

Después se vio que la frecuencia óptima teniendo en cuenta las características que buscamos es 200Hz (Anexo A).

4.1.3 Función de transferencia

El siguiente paso ha sido la caracterización de los motores, que consiste en la identificación de una función que describa la relación entre las variables de entrada y de salida del motor, esto es, entre la señal eléctrica de entrada y el sistema mecánico de rotación. Dicha función es una función de transferencia, definida por la siguiente ecuación:

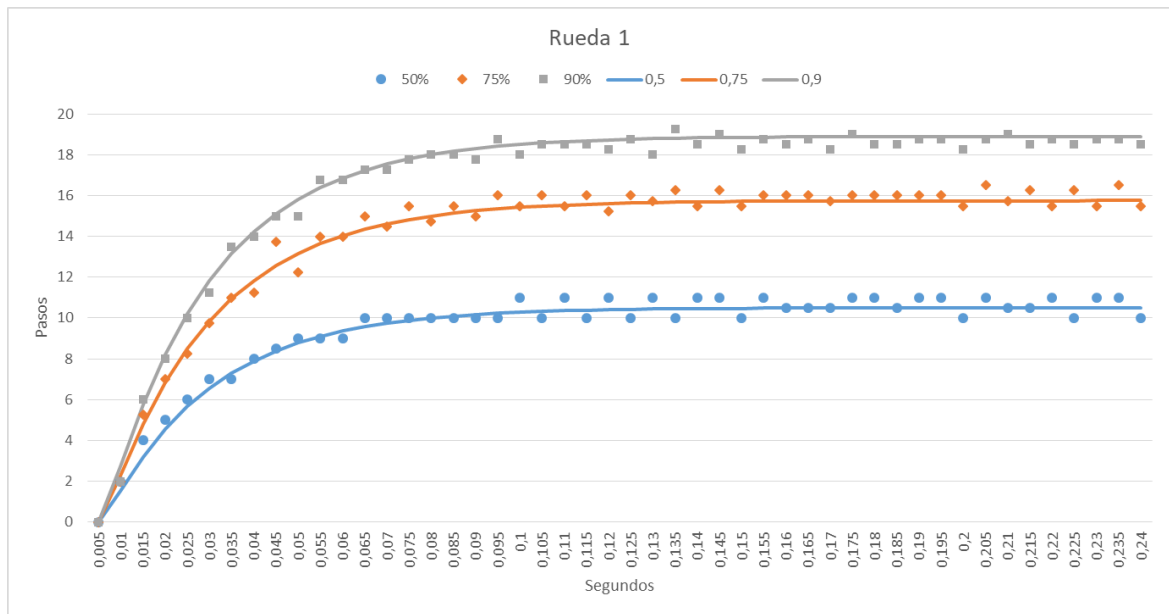
$$G(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Para determinar el valor de los coeficientes se han utilizado parte de las pruebas realizadas (Anexo B): velocidad de 0% a 50%, de 0% a 75%, de 0% a 90% (el porcentaje es respecto a la velocidad máxima posible).

Se han hallado los valores de los parámetros de la función (modelo) para ajustar su gráfica a los datos experimentales obtenidos en estas primeras pruebas. Para ello se ha utilizado la herramienta Solver de Excel con la función minimizar: minimizando el error cuadrático entre los datos experimentales y los de la función. Dado que con Solver sólo se puede minimizar una función, lo que se ha minimizado ha sido la suma total de los errores de las 3 medidas (50, 75 y 90). Cada una de estas tres medidas son el promedio de las 4 mediciones hechas para cada velocidad (50, 75 y 90).

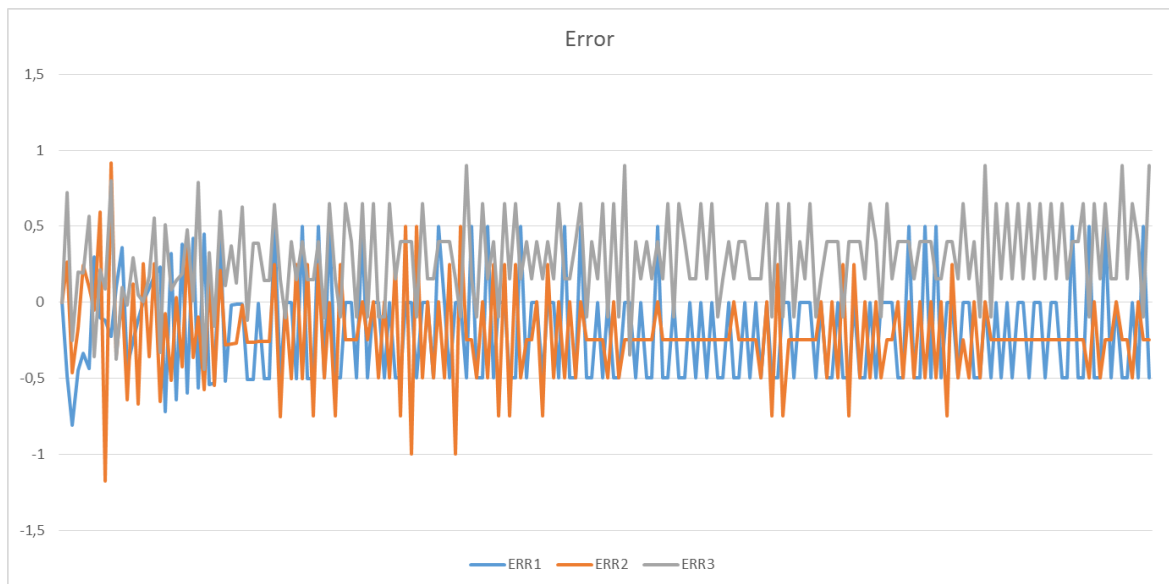
A continuación, en la Gráfica 1, se ve el dibujo de las tres velocidades de la rueda 1, gris 90%, naranja 75% y azul 50%. Los puntos son los datos a ajustar y la línea es la representación de la función que se desea ajustar, las muestras están tomadas cada 5 milisegundos, y la velocidad está en pasos. La gráfica está sesgada para una mejor visualización y no presenta todos los datos, sólo aparecen 48 muestras (0,24 segundos) mientras que se utilizan 200 muestras (1 segundo); en cambio en la Gráfica 2, donde se muestra el error de ajuste para cada velocidad de la rueda 1, sí que aparecen todos los valores de error del ajuste. El error es la diferencia entre el dato y el valor de la función en ese tiempo; descrito en la leyenda aparece *ERR1*, que corresponde al 50%, *ERR2* al 75% y *ERR3* al 90%.

En el Anexo C están las gráficas de ajuste para las 3 ruedas y sus errores, que se interpretan igual que la Gráfica 1, descrita anteriormente.



Gráfica 1: Ajuste de la gráfica de la función de la rueda 1

Sin tener en cuenta la variación en los primeros compases, donde el desajuste es mayor, podemos afirmar que el error del 50% (azul) está entre -0.5 y 0, que la mayor parte del 75% (naranja) está entre -0.75 y 0.25, y que el error del 90% (gris) está entre -0.1 y 0.6.



Gráfica 2: Error del ajuste de la función de la rueda 1

Y entonces las funciones de transferencia de las 3 ruedas quedarían así:

Para la rueda 1:

$$\frac{0,13335 z^2 - 0,101426 z - 0,0164534}{z^2 - 0,5063 z - 0,4937}$$

Para la rueda 2:

$$\frac{0,12571 z^2 - 0,129569 z + 0,0161259}{z^2 - 0,6895 z - 0,3105}$$

Para la rueda 3:

$$\frac{0,19579 z^2 - 0,154948 z + 0,0158929}{z^2 - 0,7402 z - 0,2598}$$

4.1.4 Reguladores

Una vez determinada la función de ganancia del sistema se ha utilizado Sisotool de Matlab para crear las funciones de transferencia de los reguladores.

Y simplemente ha bastado con establecer un *timer* que saltase cada 5 milisegundos y ejecutase las 3 funciones de los reguladores, uno por cada rueda.

4.2 Modelo robot

4.2.1 Adaptación del paper

Hasta aquí todo el trabajo realizado formaba parte del sistema de control de los motores, es en esta última parte, en la que se implementa el modelo cinemático del robot.

Se ha seguido el presentado en el paper *Modelado cinemático y dinámico de un robot móvil omnidireccional* [1]. En él se describe el proceso de obtención de los modelos dinámicos y cinemáticos para robots móviles de tres ruedas, y se concentra en aplicarlo a un robot móvil omnidireccional idéntico al empleado en este trabajo.

Tras entender el modelo planteado en el paper, sólo quedaba programar las funciones que realizasen los cálculos del modelo; sin embargo, en el paper no se tenía en cuenta un factor fundamental: la velocidad del motor es limitada. Esto puede aparentemente ser un detalle nimio, y en cambio, debido a la colocación y funcionamiento de las ruedas, la velocidad de cada una es vital para un correcto desplazamiento.

Para ilustrar este problema, pensemos en el siguiente ejemplo: deseamos que el robot se mueva en una dirección que requiere que las velocidades de cada rueda sean tales que tengan una relación de 3:3:1; si las velocidades asignadas son 30 pasos por segundo para las dos primeras y 10 para la tercera, y la velocidad máxima es de 10 pasos por segundo, el robot tendrá las 3 ruedas girando a la misma velocidad, por lo que estará dando vueltas.

Queda claro entonces que la puntualización es importante. De tal forma que fue preciso calcular la velocidad máxima de las ruedas para programar una función que se encargara de transformar estas magnitudes obtenidas por las ecuaciones cinemáticas, en consignas adaptadas a la realidad del robot.

Esta adaptación consiste en procesar las velocidades obtenidas del modelo de tal manera, que si alguna supera la velocidad angular máxima de las ruedas entonces se reducen todas proporcionalmente a esa escala, correspondiendo la más alta con la máxima (10,123). En la Figure 4 aparece la velocidad de cada rueda antes y después de aplicar esta normalización.

Otro reajuste que hubo que hacer fue con la matriz del modelo, pues el sistema de coordenadas que utiliza el modelo del paper no es el usual. En los dibujos Figure 4 y Figure 5 se ven ambos sistemas de coordenadas.

Fue por ello que las ecuaciones que se derivan de operar el modelo con la matriz jacobiana, hubo que cambiarlas. La operación es simplemente una rotación de 90 grados de los vectores X e Y, y las ecuaciones pasan de ser las escritas en (1) a ser las escritas en (2).

$$\begin{cases} \omega_1 = (x \cdot \sin \Theta_1 - y \cdot \cos \Theta_1 - L \cdot z) \cdot \frac{1}{r} \\ \omega_2 = (x \cdot \cos \Theta_2 + y \cdot \sin \Theta_2 - L \cdot z) \cdot \frac{1}{r} \\ \omega_3 = (x \cdot \cos \Theta_3 + y \cdot \sin \Theta_3 - L \cdot z) \cdot \frac{1}{r} \end{cases} \quad (1)$$

$$\begin{cases} \omega_1 = (x \cdot \cos \Theta_1 + y \cdot \sin \Theta_1 - L \cdot z) \cdot \frac{1}{r} \\ \omega_2 = (-x \cdot \sin \Theta_2 + y \cdot \cos \Theta_2 - L \cdot z) \cdot \frac{1}{r} \\ \omega_3 = (-x \cdot \sin \Theta_3 - y \cdot \cos \Theta_3 - L \cdot z) \cdot \frac{1}{r} \end{cases} \quad (2)$$

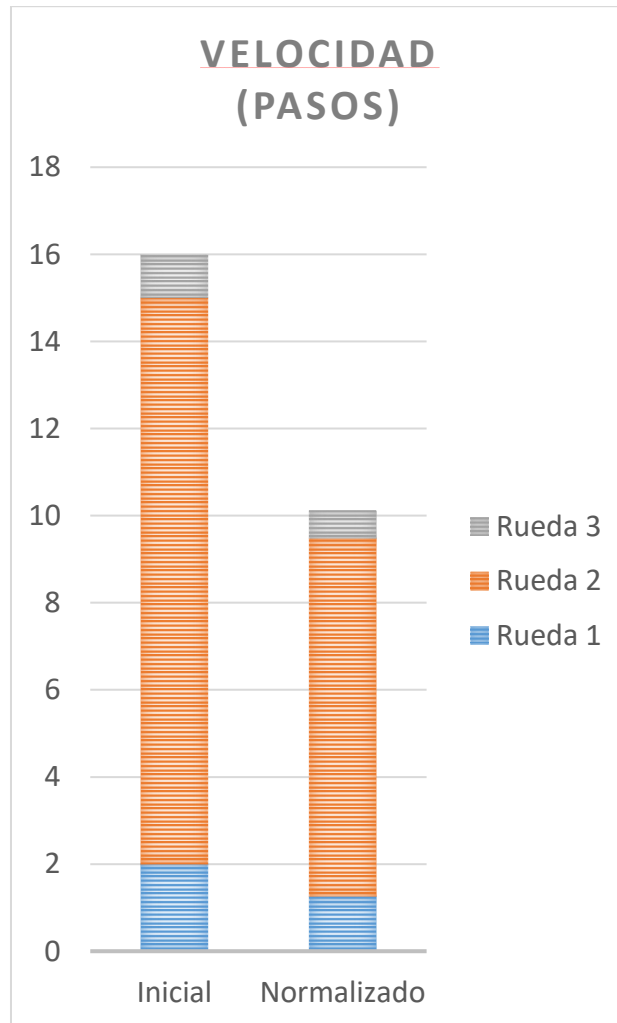


Figure 3: Comparación de las velocidades antes y después de normalizar

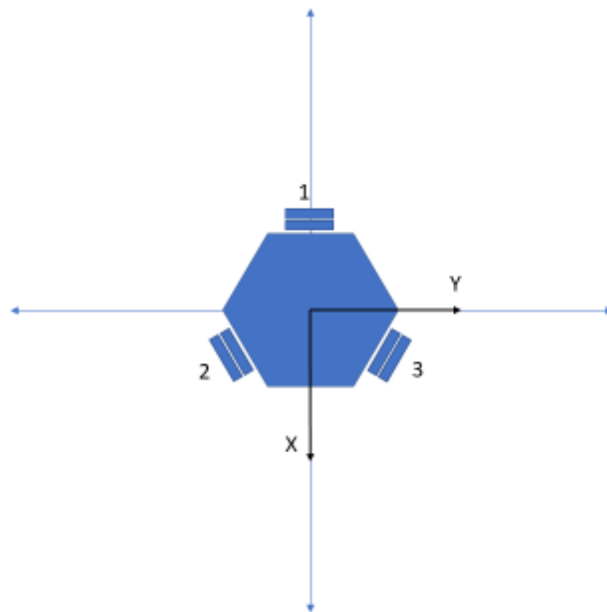


Figure 4: Sistema de coordenadas del paper

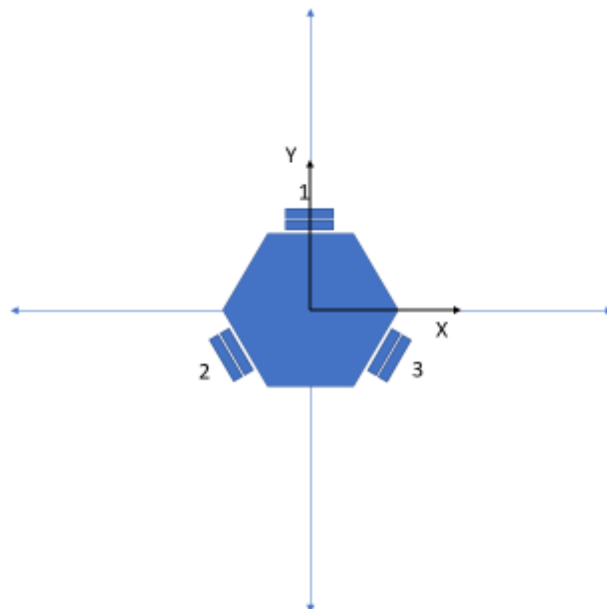


Figure 5: Sistema de coordenadas deseado

5 Integración, pruebas y resultados

1.1.Integración

La integración inicialmente ha consistido simplemente en ir añadiendo cada parte nueva que se iba desarrollando, al todo. Pero una vez finalizado el desarrollo, ha sido necesaria una reestructuración del código para una mayor legibilidad y comprensión, así como para facilitar la incorporación de futuros cambios y mejoras.

1.2.Pruebas y resultados

Para comprobar el correcto funcionamiento se diseñaron pruebas con los siguientes vectores de posición:

- 1,3
- 1,4
- 2,3
- 2,4
- 3,3
- 3,2
- 3,1
- 4,1
- 4,2
- 0,1
- 1,0
- -1,0
- 0,-1

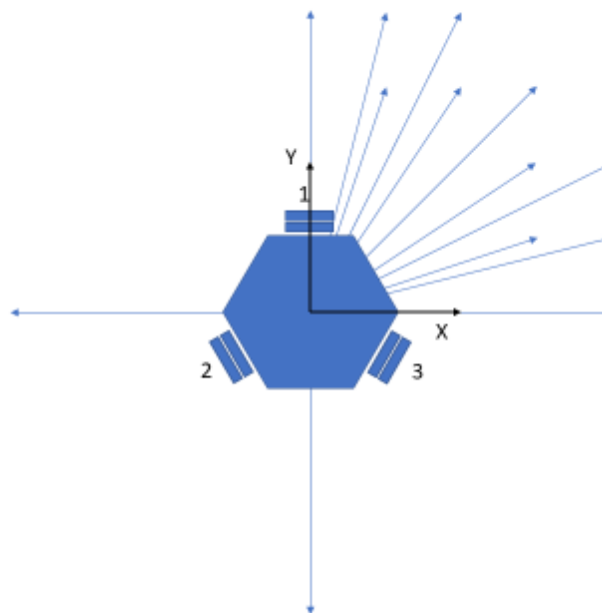


Figure 6: Disposición del robot en las pruebas

El objetivo era posicionar al robot en el suelo, y que desde en el centro de coordenadas, que es siempre su centro geométrico, se moviera a los puntos antes mencionados. En la figura 6 está representado el robot colocado en la posición inicial, y los lugares a donde está pensado que se mueva, indicados mediante flechas.

Y en la Ilustración 5-I se ha reproducido con exactitud los vectores comandados al robot. Cada vector es de un color para compararlos con los vectores correspondientes de los resultados obtenidos.

Tras la realización de las primeras pruebas se observó en los resultados que era necesaria una modificación. Como se puede ver en la Ilustración 5-II, las longitudes de las distancias recorridas no son consistentes con las unidades comandadas. Esto se debe a que el tiempo de recorrido no estaba bien constituido.

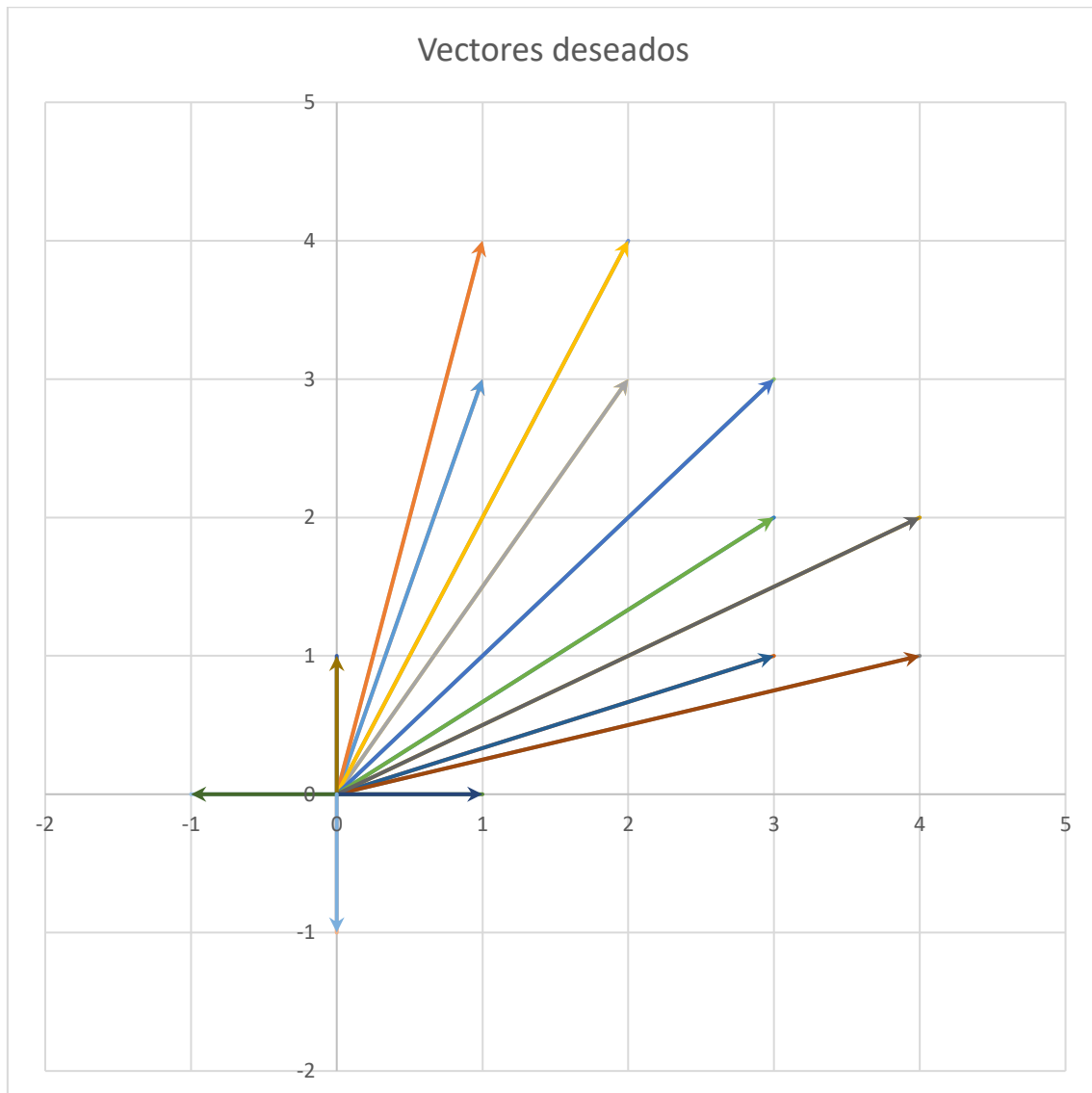


Ilustración 5-I: Vectores que indican las posiciones deseadas

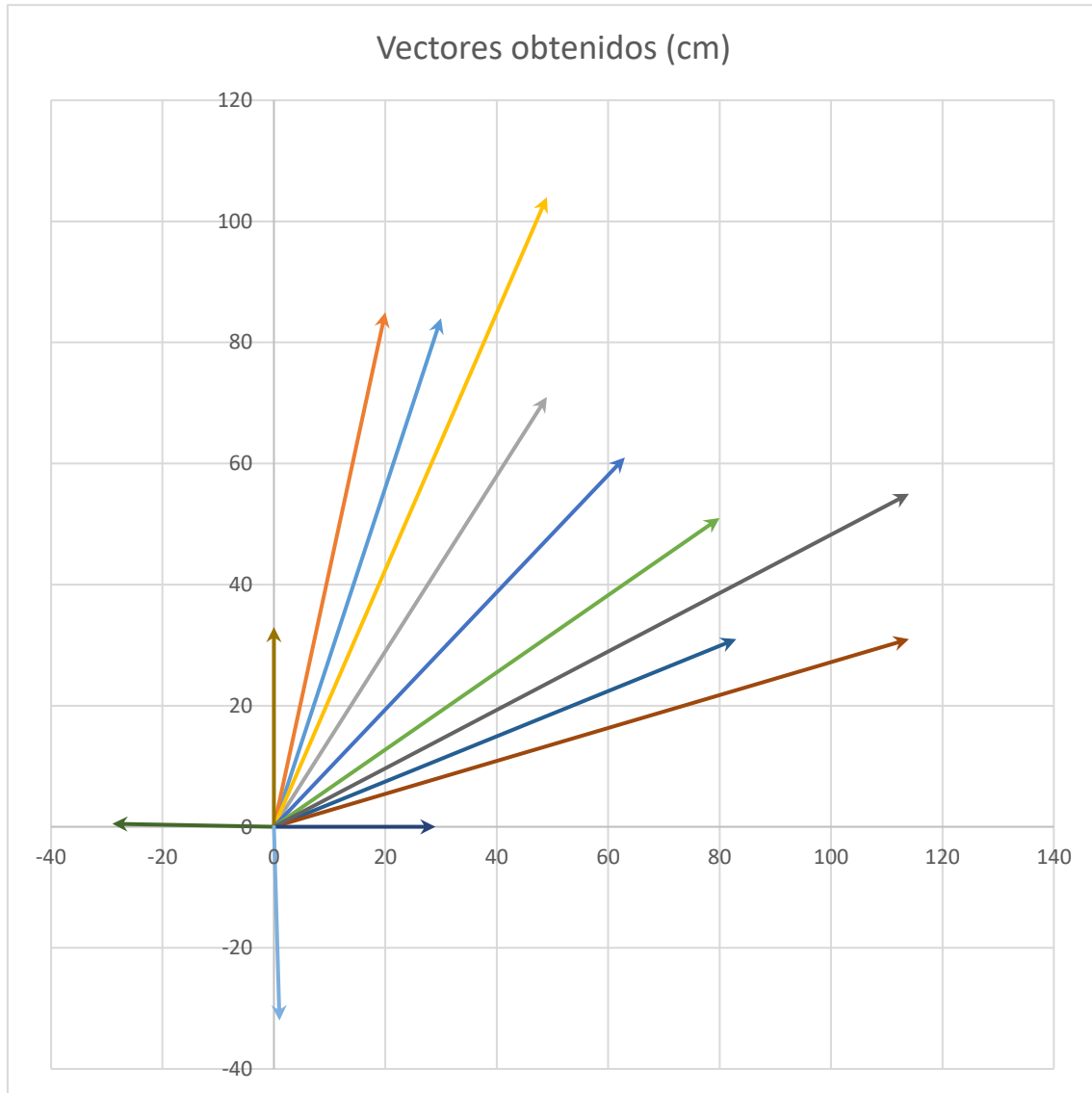


Ilustración 5-II: Vectores de posición obtenidos en las primeras pruebas

Así, el cambio que se introdujo tras la primera batería de pruebas, consiste en calcular el módulo de la posición final y convertirlo en el número de segundos que el robot está en movimiento hasta que se detiene. Es decir, que si el vector de posición es (4,3), el tiempo que el robot se está moviendo son 5 segundos ($5 = \sqrt{4^2 + 3^2}$).

Realizando de nuevo la misma batería de pruebas, los resultados esta vez mejoraron considerablemente, obteniendo un alto grado de similitud con los resultados deseados. Tras el cambio los resultados fueron los de la Ilustración 5-III.

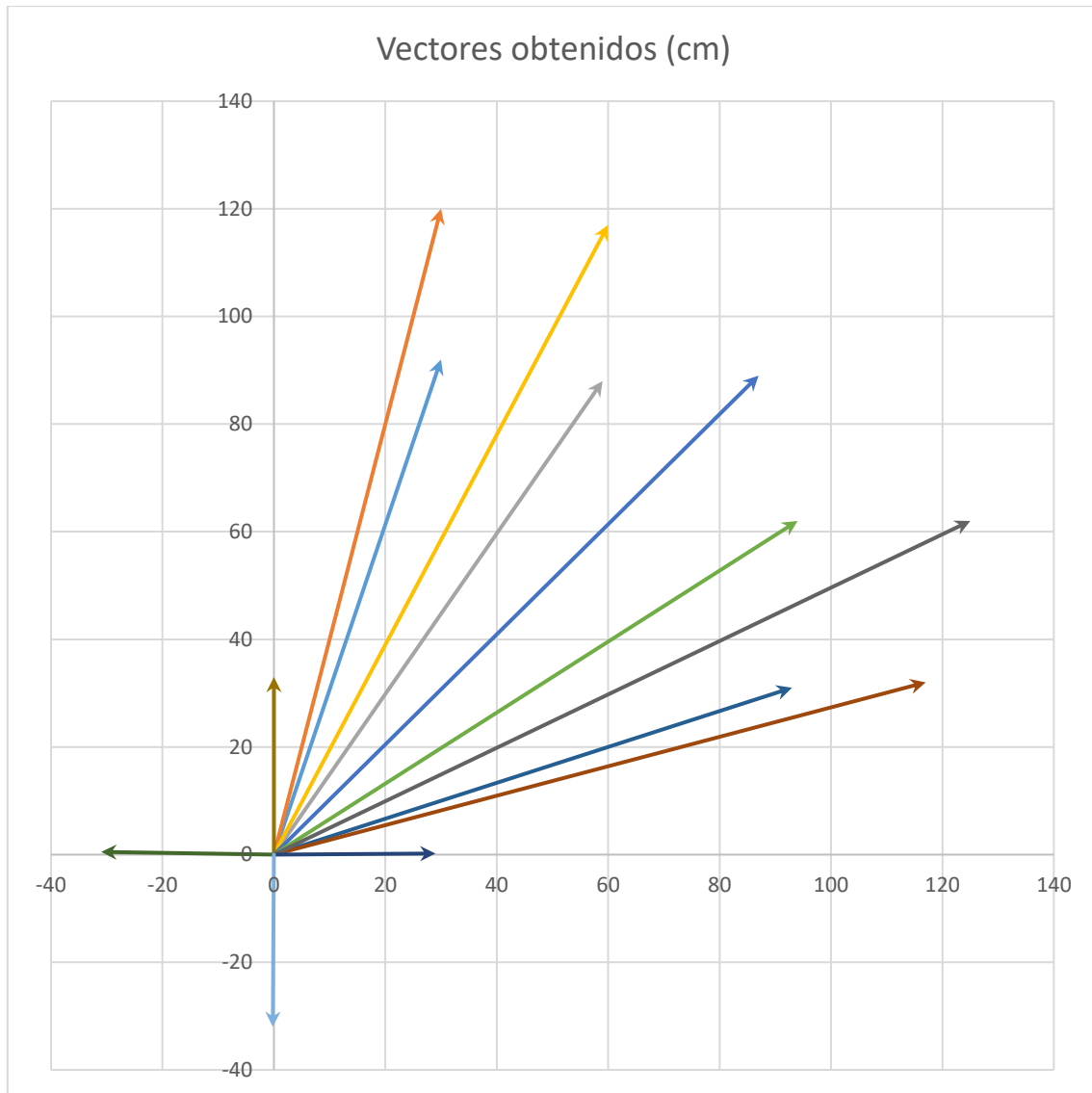


Ilustración 5-III: Vectores de posición obtenidos tras la modificación

Se puede ver que el vector unidad son 30 cm. Siguiendo este criterio podemos medir el error de precisión como la longitud del vector diferencia entre el vector comandado y el medido, que no es más que el módulo de la resta entre los dos vectores. En la Ilustración 5-IV se representa este error para cada vector. El error no tiene unidades porque se ha calculado como la diferencia entre los vectores coordenadas deseados y obtenidos, pero previamente dividiendo entre 30 cm el vector medido, dejándolo así sin unidades; por tanto también pueden ser entendidos como porcentajes de la unidad.

Por ejemplo, del vector (1,0) enviado al robot, hemos obtenido el vector (29,0) en cm, que dividido entre 30 cm queda el vector (0.96667, 0), y la diferencia entre ambos es (0.03333, 0), que tiene longitud $\sqrt{0.03333^2 + 0^2} = 0.03333$.

También podemos decir que tiene un grado de precisión medio del 90.4511465%, que se obtiene si hacemos la media de todos los errores obtenidos (0.095488535).

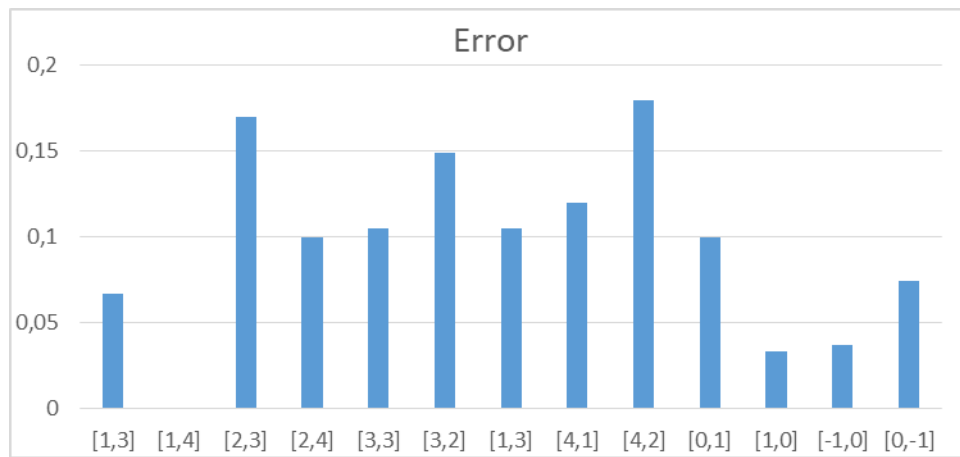


Ilustración 5-IIIV : Error de precisión del movimiento

6 Conclusiones y trabajo futuro

Este trabajo no era de investigación, en él se buscaba implementar un modelo ya conocido para el control del movimiento de robots omnidireccionales de tres ruedas, y por tanto la solución esperada estaba muy bien definida. Así, el desarrollo del trabajo ha sido satisfactorio, llegando a implementar el sistema de direccionamiento para el robot.

Como se puede apreciar en las pruebas finales el sistema no tiene precisión milimétrica, pero se aproxima bastante, su error medio no supera el 10%. Esto es debido principalmente a dos factores:

- El primer factor es que el regulador no es exacto, y los pasos por segundo a los que se mueve es una aproximación de la velocidad ideal de cada rueda, limitada por la unidad del paso.
- El otro factor es el sistema que controla el avance del robot. La posición se calcula por el movimiento de las ruedas, que pueden patinar; es decir, que el avance registrado del robot puede no corresponderse con el real.

En las pruebas se han podido captar las evidencias de este segundo factor: cuando las ruedas dejaban de rotar, el robot no se frenaba en seco, sino que resbalaba.

El primer objetivo era que el robot se desplazara al lugar indicado. Para eso no se pretendía desarrollar un sistema con el que comunicarse en tiempo real mediante Bluetooth o WiFi para mandar al robot las instrucciones de movimiento. Eso sería posterior en caso de que hubiese tiempo para hacerlo, para simplificar el desarrollo, inicialmente se introducirían las coordenadas en el código. Por tanto, una mejora interesante sería implementar este servicio que permitiría guiar al robot a lo largo de un trayecto, porque actualmente el sistema permite sólo la introducción de las coordenadas finales, la trayectoria es directa y no considera los obstáculos.

Por este último motivo, otra opción para evolucionarlo es utilizar los sensores para que pueda percibir el entorno y saber si la ruta está libre de obstáculos, o por el contrario debe trazar otro recorrido. Que a su vez también podría ser calculada por otro modelo que habría que implementar, o simplemente que se parase y esperase a que se retirara el objeto.

Sin relación con la cuestión anterior, también se podría tener en cuenta, en un futuro, la rotación del robot, que aunque está implementada, para la consecución de nuestros objetivos no se ha utilizado para nada, y por tanto no se han hecho pruebas de la correcta rotación.

Pero el trabajo que quizá convenga hacer antes, es calibrar la precisión del movimiento. Buscar soluciones para minimizar el error derivado de los dos factores mencionados anteriormente. Esto pasaría por refinar los valores obtenidos para la función de transferencia de cada rueda, e implementar un regulador más depurado que dé resultados más precisos.

Referencias

- [1] Muñoz-Martínez, Víctor & Gil-Gómez, G & García, Alfonso. (2018). Modelado cinemático y dinámico de un robot móvil omni-direccional.
- [2] A. Ollero. Robótica: Manipuladores y robots móviles. Marcombo, 2001.
- [3] Gracia Calandín, L.I. (2006). Modelado cinemático y control de robots móviles con ruedas [Tesis doctoral]. Universitat Politècnica de Valencia.
- [4] Efrén Barrero Páez, Luis & Felipe Villegas Caballero, Andrés & Carolina Gómez Gómez, Diana. (2014). Robot transportador y distribuidor de objetos según su peso. Redes de Ingeniería. 5. 5. 10.14483/2248762X.6356.
- [5] Leonardo E. Solaque Guzmán, Manuel A. Molina Villa, Edgar L. Rodríguez Vásquez. (2014). Seguimiento de trayectorias con un robot móvil de configuración diferencial. Ing. USBMed, Vol. 5, No. 1, pp. 26-34. ISSN: 2027-5846.
- [6] Bermúdez, Giovanni. (2003). Modelamiento cinemático y odométrico de robots móviles: aspectos matemáticos. Tecnura.
- [7] Ríos G., Luís Hernando, Bueno L., Maximiliano. Modelo matemático para un robot móvil. Scientia Et Technica, vol. XIV, núm. 38, junio, 2008, pp. 13-18. ISSN 0122-1701.
- [8] Guerrero Castellanos, J.F. .Seguimiento de trayectorias de un robot móvil (3,0) mediante control acotado. Revista Iberoamericana de Automática e Informática industrial, [S.l.], v. 11, n. 4, p. 426-434, oct. 2014. ISSN 1697-7920.
- [9] Una Introducción a los Robots Móviles, Il Bambino, 2008. Disponible en:
http://www.aadeca.org/publicaciones/cponline/monografia_robot_movil.pdf
- [10] Tutorial de sensores de posición. Disponible en:
https://www.electronics-tutorials.ws/io/io_2.html

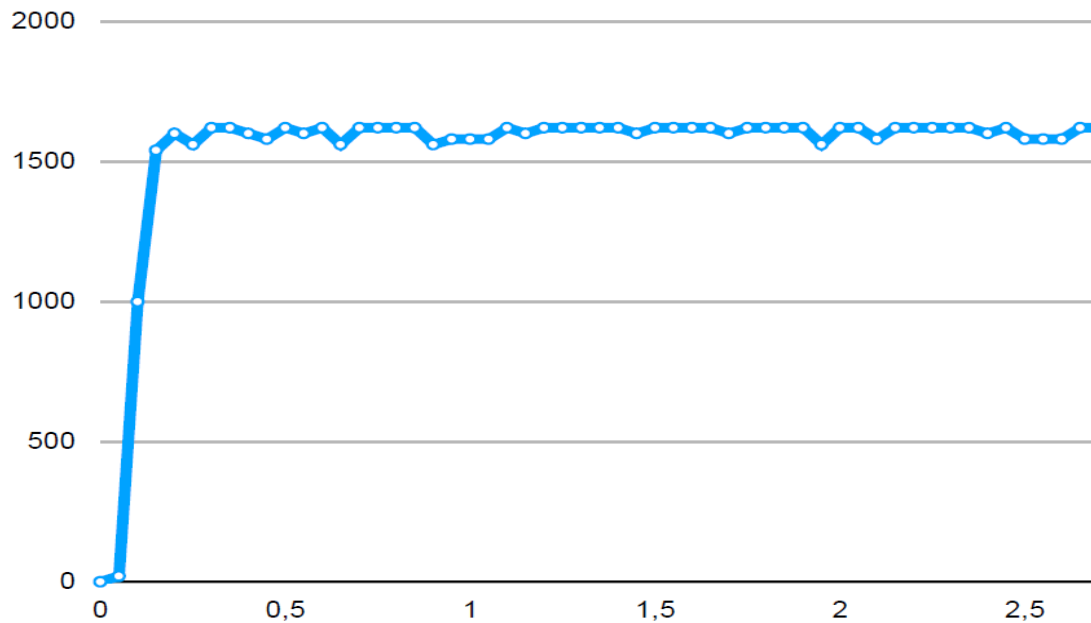
Glosario

API	Application Programming Interface
FDT	Función De Traslación

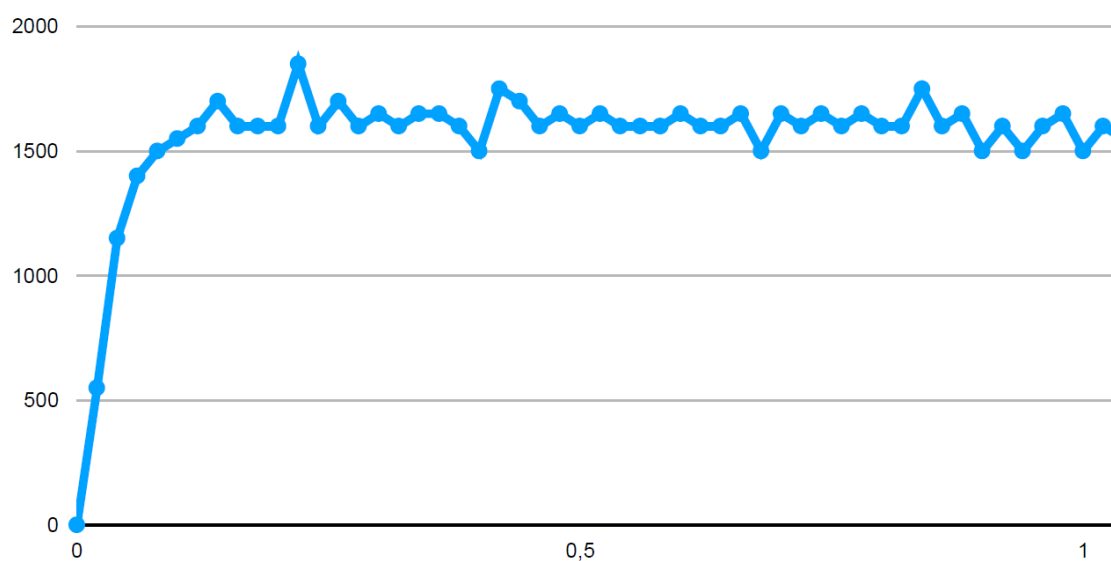
Anexos

A Pruebas con varias frecuencias

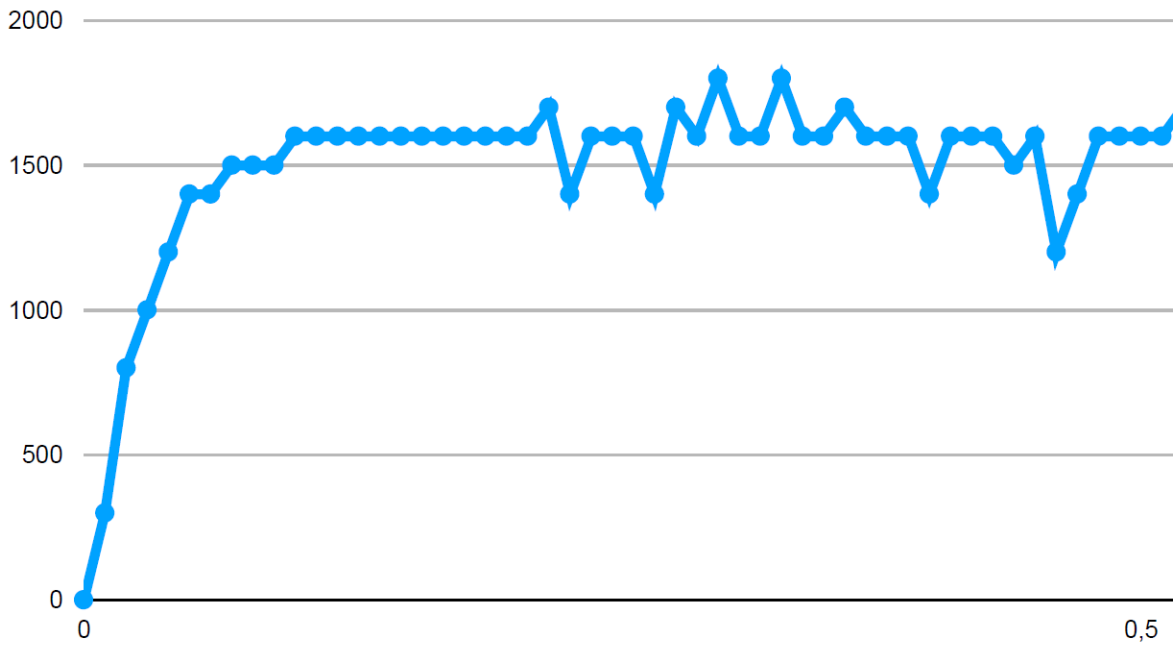
La diferencia en la frecuencia de muestreo da lugar a distintas observaciones del comportamiento de la velocidad de las ruedas. Las siguientes gráficas lo muestran para ciertas frecuencias.



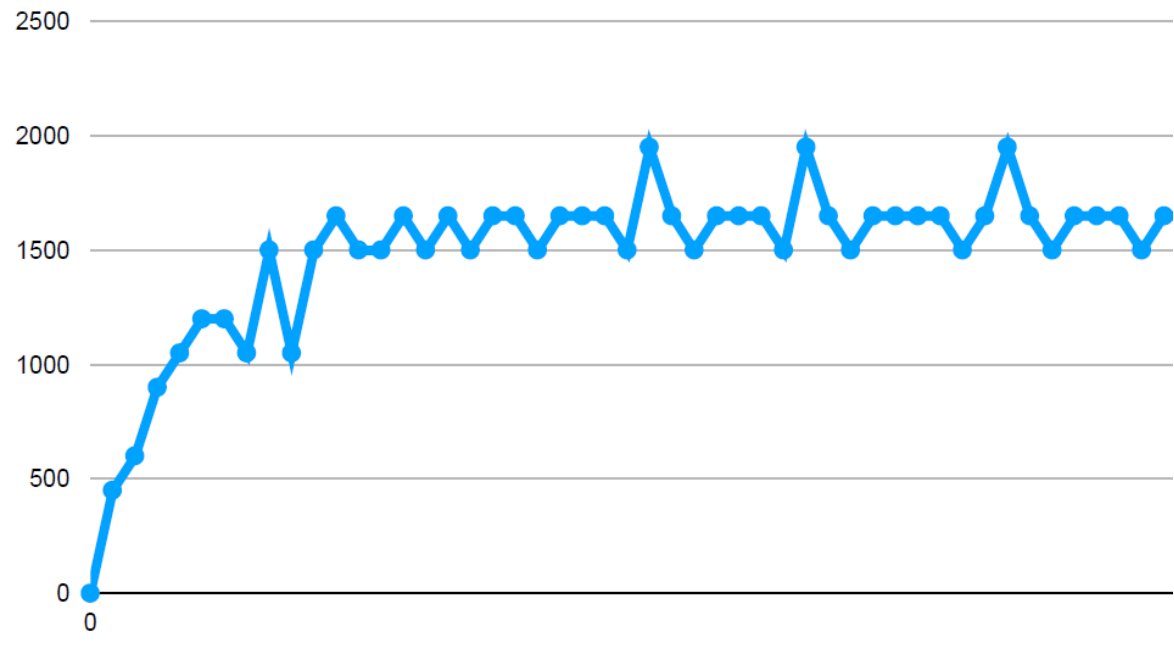
Gráfica 3: 20 Hz



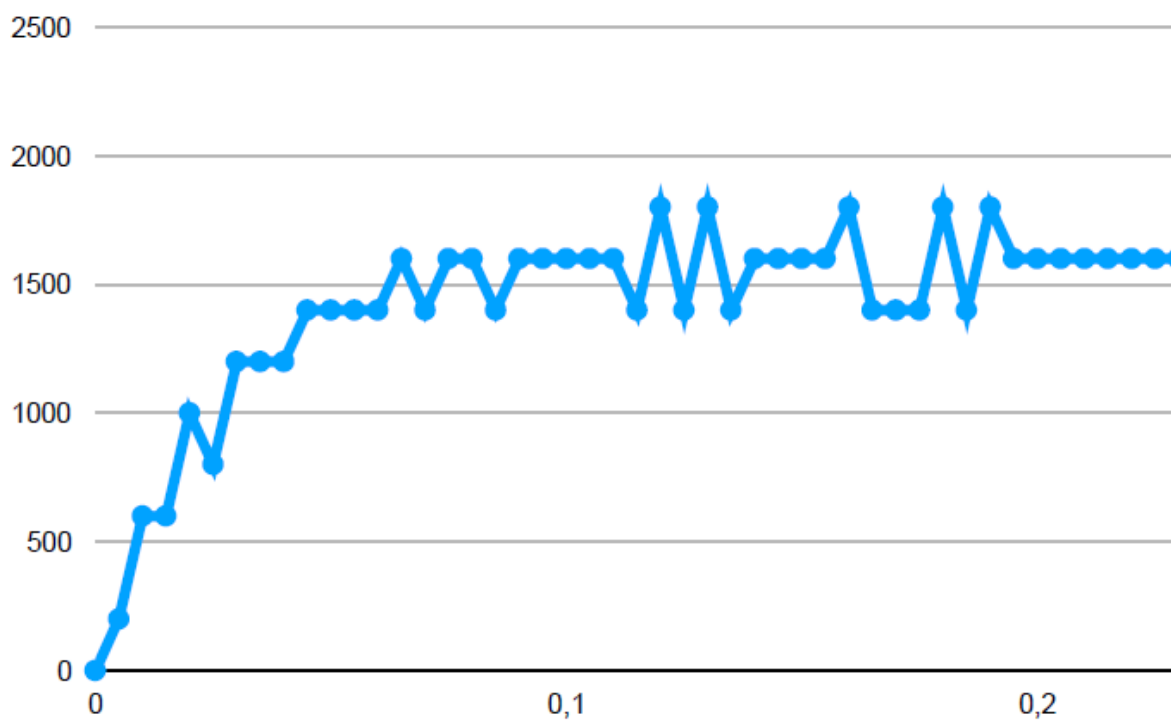
Gráfica 4: 50 Hz



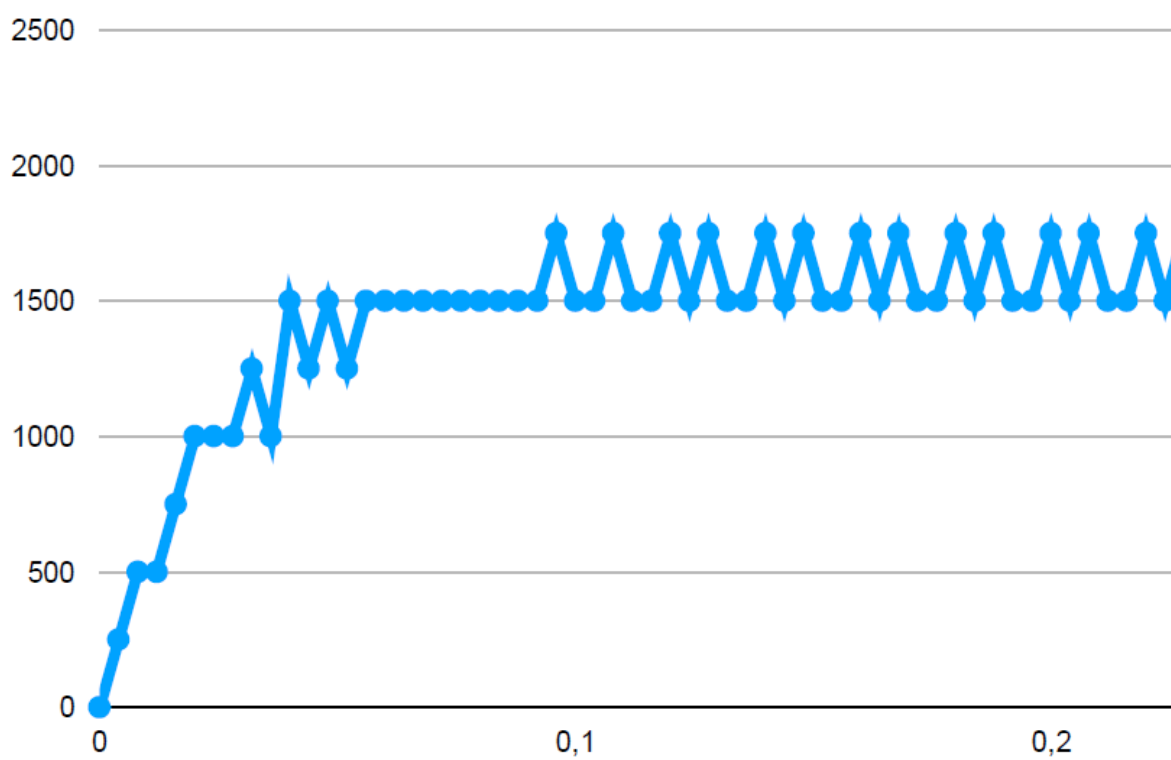
Gráfica 5: 100 Hz



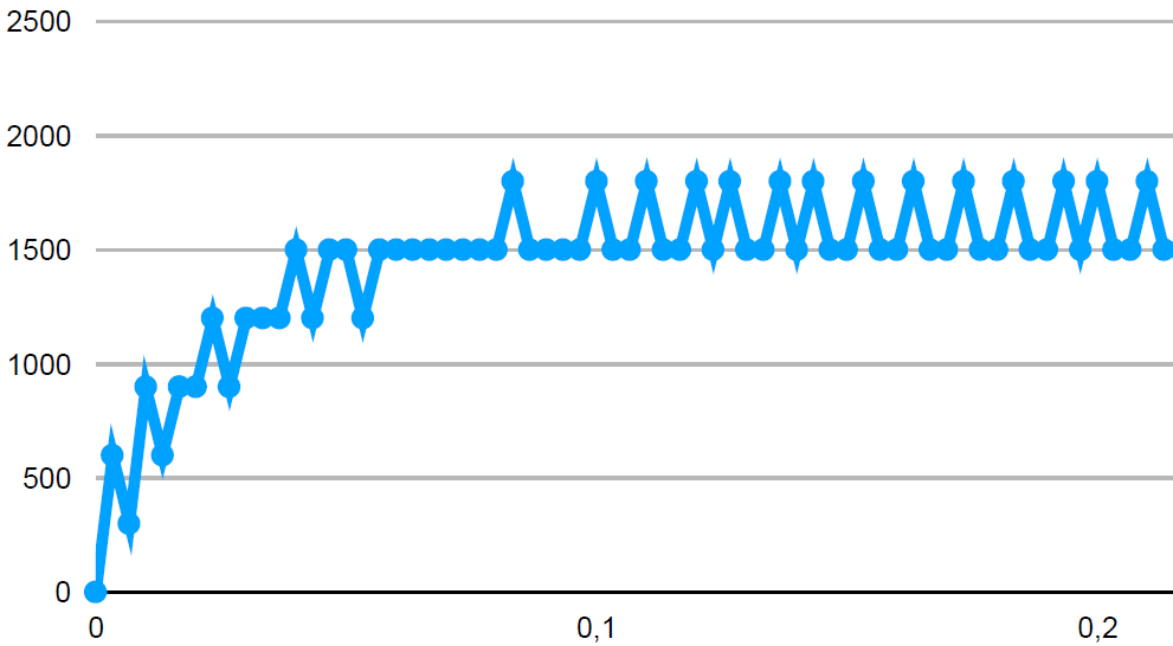
Gráfica 6: 150 Hz



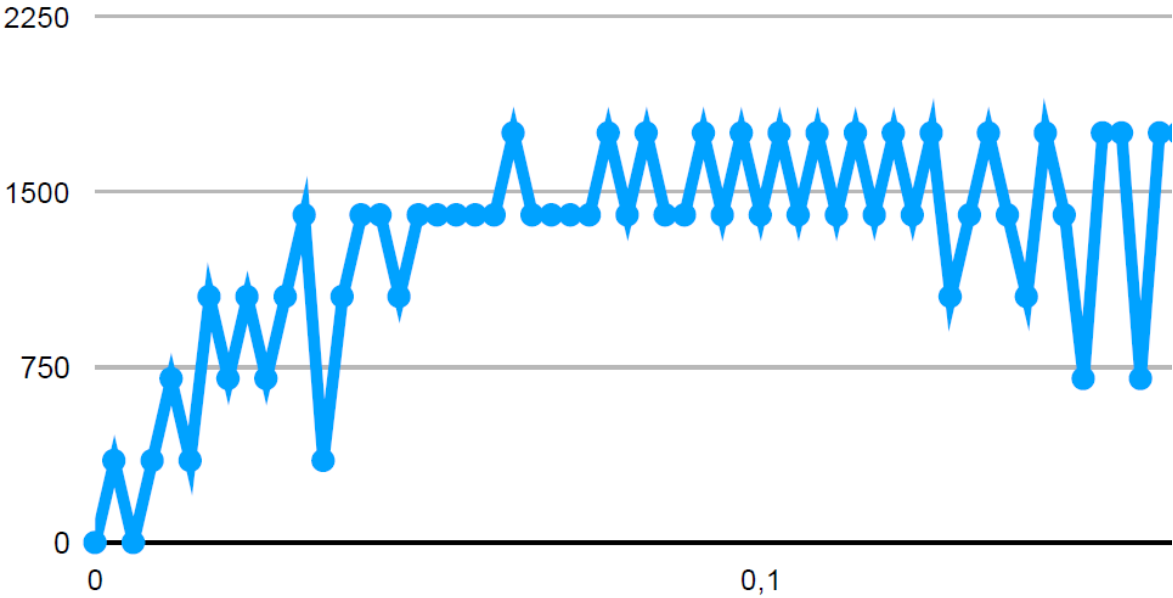
Gráfica 7: 200 0Hz



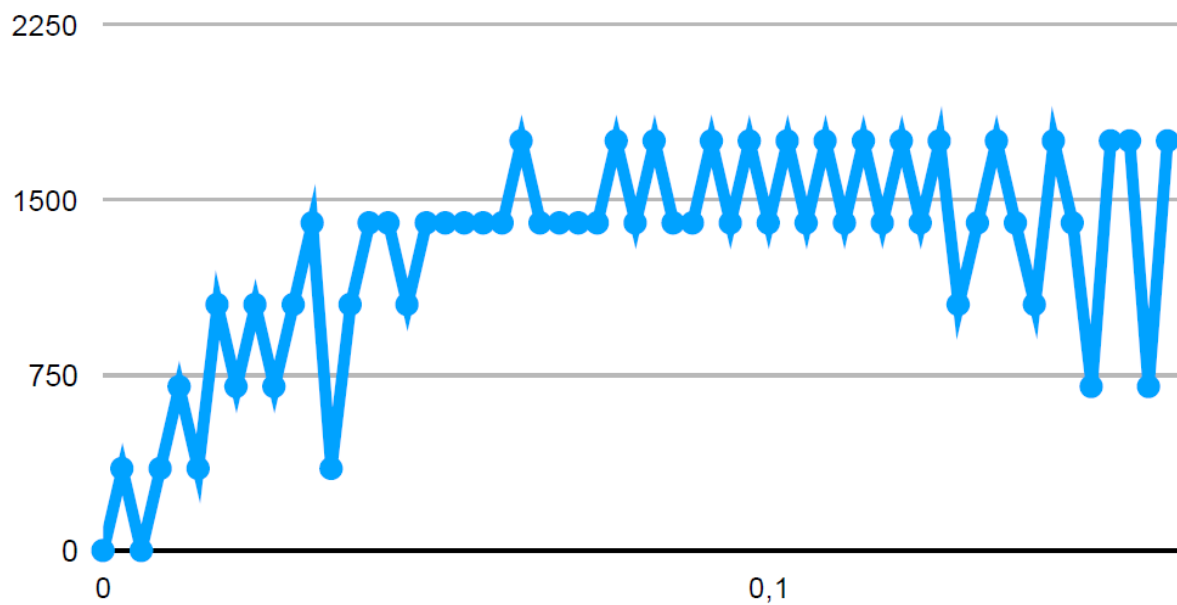
Gráfica 8: 250 Hz



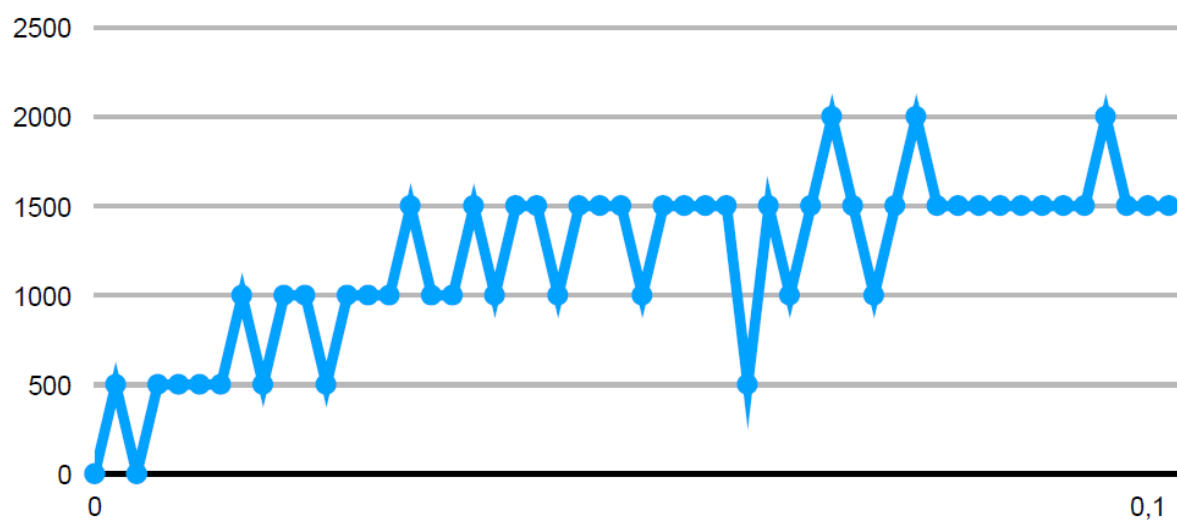
Gráfica 9: 300 Hz



Gráfica 10: 350 Hz



Gráfica 11: 400 Hz



Gráfica 12: 500 Hz

B Medidas ruedas

Las variables de la función de transferencia del regulador son las que más ajustan el modelo creado al comportamiento real de las ruedas. Para ello se han tomado una serie de muestras que sirvan para comparar con el modelo. A continuación se ilustran las muestras en forma de tabla (solamente con las 40 primeras muestras) y con gráficas.

B.1 Tablas

(1) Rueda 1

RUEDA 1											
0 -50 (%)				0 - 75 (%)				0 - 90 (%)			
0	0	0	0	0	0	0	0	0	0	0	0
2	2	2	2	2	2	2	2	2	2	2	2
4	4	4	4	5	5	6	5	6	6	6	6
5	5	5	5	7	7	7	7	8	8	8	8
6	6	5	6	9	8	8	8	10	10	10	10
7	6	7	7	9	10	10	10	11	11	11	12
7	7	7	7	11	11	11	11	13	14	14	13
8	8	8	8	11	11	12	11	14	14	14	14
8	9	8	9	14	14	13	14	15	15	15	15
9	9	10	9	12	12	13	12	15	15	15	15
10	9	9	9	14	14	14	14	16	17	17	17
9	9	9	9	14	14	14	14	17	16	17	17
10	10	10	10	15	15	15	15	17	18	17	17
10	10	10	10	15	14	14	15	18	17	17	17
10	10	9	10	15	16	16	15	17	18	18	18
10	10	11	10	15	15	14	15	18	18	18	18
10	10	10	10	15	15	16	16	18	18	18	18
10	10	10	10	15	15	15	15	18	18	18	17
10	11	10	10	16	16	16	16	19	19	18	19
11	10	11	11	16	16	15	15	18	18	19	17
10	10	10	10	16	16	16	16	18	19	18	19
10	11	11	11	15	15	16	16	19	18	19	18
11	10	10	10	16	16	16	16	19	19	18	18
10	11	11	11	15	16	15	15	18	18	19	18
11	10	10	10	16	16	16	16	19	19	19	18
10	11	11	11	16	15	16	16	18	18	18	18
11	10	10	10	16	17	16	16	19	20	19	19
10	11	11	11	16	15	16	15	19	18	19	18
11	11	11	11	16	16	16	17	19	19	19	19
10	10	10	10	16	16	15	15	18	19	18	18
11	11	11	11	16	16	16	16	19	19	19	18
11	10	10	11	16	16	16	16	19	18	19	18
10	11	11	10	16	16	16	16	18	19	19	19
11	10	10	11	15	16	16	16	19	18	18	18
10	11	11	11	16	16	16	16	19	19	19	19

(2) Rueda 2

RUEDA 2											
0 -50 (%)				0 - 75 (%)				0 - 90 (%)			
0	0	0	0	0	0	0	0	0	0	0	0
2	2	2	2	2	2	2	2	2	2	2	2
4	4	4	4	5	6	5	6	6	6	6	6
5	5	5	5	7	7	7	7	8	8	8	8
6	6	6	6	9	8	9	9	9	10	10	10
6	6	6	7	9	10	9	9	12	12	11	11
7	7	8	7	11	11	11	11	13	13	13	13
8	8	8	8	12	12	12	12	14	14	14	14
9	9	8	9	13	13	13	13	15	15	15	15
8	8	9	8	13	13	13	13	16	16	16	16
10	10	10	10	14	14	14	14	16	17	16	17
9	9	9	9	14	14	14	14	17	16	17	16
10	10	10	10	15	15	15	15	17	18	17	18
10	10	10	10	14	15	14	15	18	17	18	17
10	10	10	10	16	15	16	15	18	18	18	18
10	10	10	10	14	15	15	15	17	18	17	18
10	10	10	11	16	16	15	16	19	18	19	18
10	11	11	10	15	15	15	15	18	18	18	18
11	10	10	11	16	16	16	16	18	18	19	18
10	11	11	10	15	15	16	15	18	19	18	18
11	11	10	11	16	16	16	16	19	18	19	19
11	10	11	10	15	16	16	16	18	19	18	18
10	11	11	11	17	16	16	16	19	18	19	19
11	10	11	11	15	15	16	16	18	18	19	18
11	11	10	10	16	16	16	16	19	19	18	19
10	10	11	11	16	16	16	16	19	19	19	18
11	11	11	11	16	16	16	16	18	18	19	18
11	11	10	10	15	15	16	16	18	19	18	19
11	11	11	11	16	16	16	16	19	18	19	19
11	11	11	11	16	16	16	16	19	19	19	18
10	10	11	10	16	16	16	16	18	18	18	19
11	11	10	11	16	16	16	16	19	19	19	18
10	11	11	11	16	16	16	16	18	18	19	19
11	10	11	11	15	16	16	16	19	19	18	19
11	11	11	11	16	16	17	16	19	18	19	18
11	11	10	11	16	15	15	16	18	18	18	19
10	11	11	11	16	17	17	16	19	19	19	19
11	11	11	10	16	15	16	16	18	18	18	18
11	10	11	11	16	16	16	16	19	19	19	19
11	11	11	11	15	16	16	16	18	18	19	19
10	11	10	11	17	16	16	16	19	19	18	19

(3) Rueda 3

RUEDA 3											
0 -50 (%)				0 - 75 (%)				0 - 90 (%)			
0	0	0	0	0	0	0	0	0	0	0	0
2	2	2	2	2	1	2	2	2	2	2	2
3	3	4	4	5	6	5	6	6	6	6	6
5	5	5	5	7	6	7	6	8	7	8	8
5	5	5	5	8	8	8	9	9	10	10	9
7	7	7	7	9	10	10	9	11	11	11	11
7	7	7	7	11	10	10	11	13	13	13	13
8	8	8	8	12	12	12	11	14	14	13	14
8	8	8	8	12	12	12	13	14	15	15	15
9	9	9	9	13	13	13	13	15	15	15	15
9	9	9	9	14	13	14	14	17	16	16	16
9	9	9	9	13	14	13	13	16	16	17	17
9	9	9	9	15	15	15	15	17	17	17	17
10	10	10	10	14	14	14	14	17	18	17	17
10	10	10	10	15	15	15	15	18	17	18	18
9	10	10	9	15	14	15	15	17	18	17	17
11	10	10	10	15	16	15	15	18	17	18	18
10	9	10	11	16	15	15	15	18	18	18	18
10	11	10	10	15	15	16	15	18	18	18	18
10	10	10	10	15	15	15	15	18	18	18	18
10	10	11	10	16	16	15	16	18	19	18	18
11	10	10	11	15	15	16	15	18	18	18	18
10	11	10	10	16	16	15	16	19	18	18	19
10	10	11	11	15	15	16	15	18	19	18	18
11	11	10	10	16	16	16	16	18	18	18	18
10	10	10	10	16	16	15	16	19	18	19	18
10	10	11	11	16	16	16	16	18	19	18	19
11	11	10	10	15	15	15	15	18	18	18	18
11	11	11	10	16	16	16	16	19	19	19	19
10	10	10	11	15	15	16	15	18	18	18	18
10	11	11	10	16	16	16	16	19	18	18	19
11	10	10	11	16	16	15	16	18	18	18	18
11	11	11	10	16	15	16	16	19	19	19	18
10	10	10	11	15	16	16	16	18	18	18	19
10	10	10	10	16	16	15	16	19	19	18	18
11	11	11	11	15	15	16	15	18	18	19	18
10	10	10	11	17	16	16	16	19	18	18	19
11	11	11	10	15	16	15	15	18	18	19	18
10	10	10	11	16	16	16	16	19	19	18	19
11	11	11	10	16	15	16	16	18	18	18	18
10	10	10	11	16	16	16	16	19	19	19	18

B.2 Gráficas

(1) Rueda 1



Gráfica 14: Rueda 1, Vel. 50, 1ª



Gráfica 15: Rueda 1, Vel. 50, 2ª

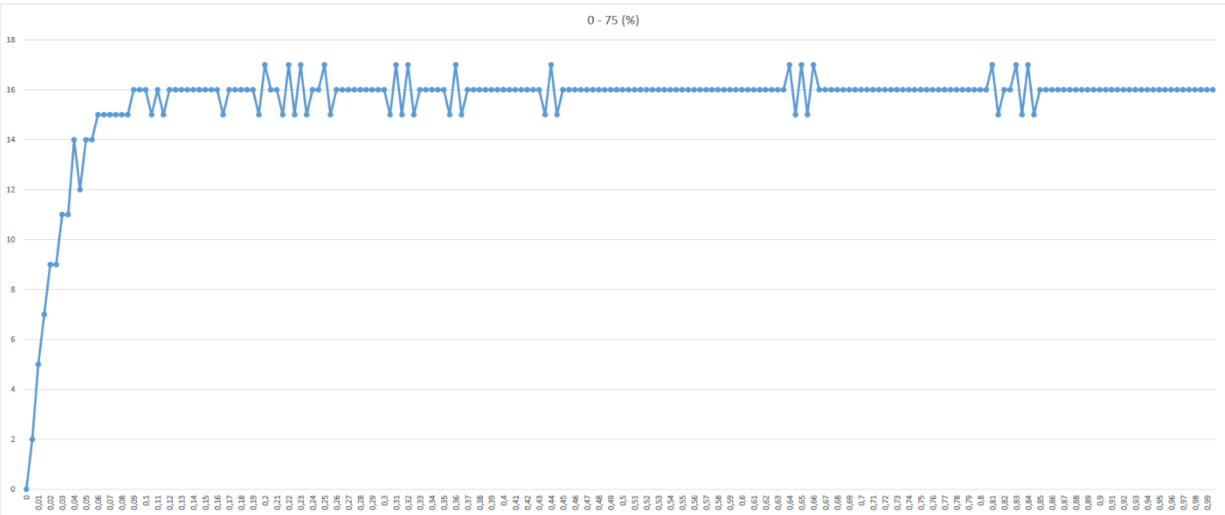


Gráfica 13: Rueda 1, Vel. 50, 3ª

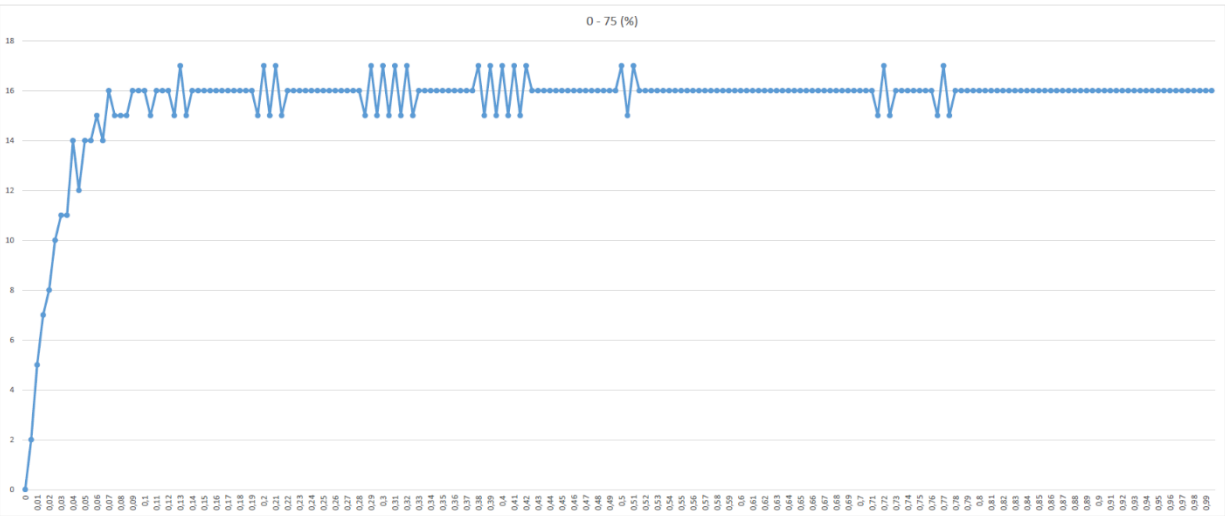
Presupuesto



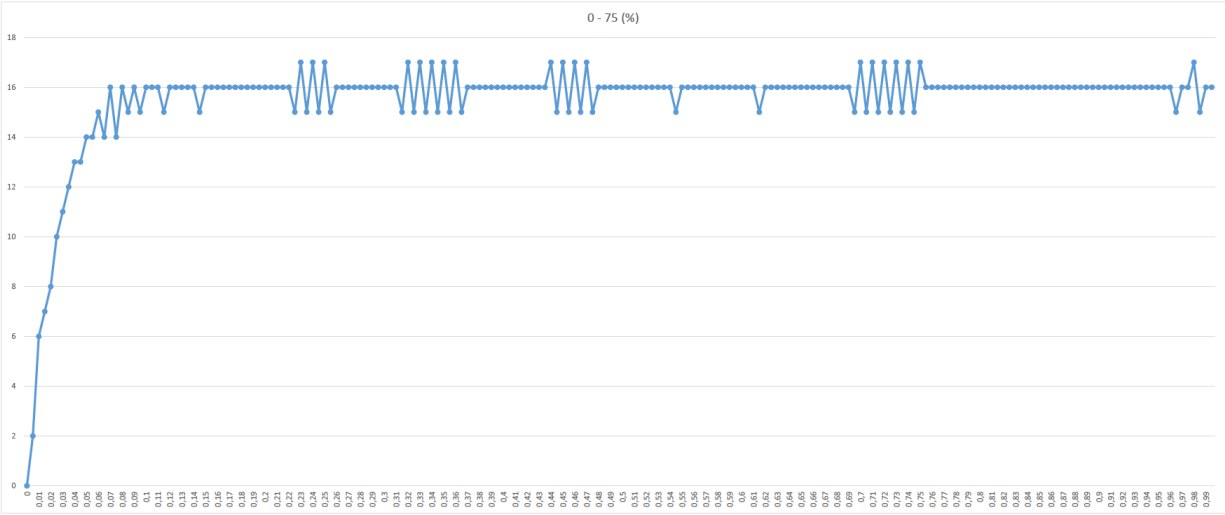
Gráfica 17: Rueda 1, Vel. 50, 4ª



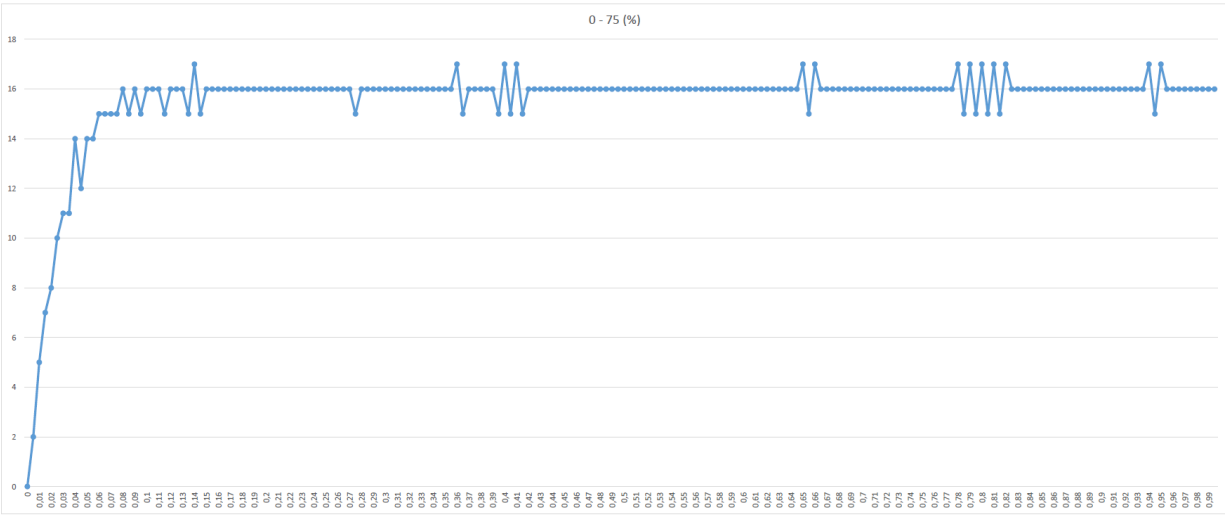
Gráfica 18: Rueda 1, Vel. 75, 1ª



Gráfica 16: Rueda 1, Vel. 75, 2ª



Gráfica 19: Rueda 1, Vel. 75, 13



Gráfica 21: Rueda 1, Vel. 75, 4ª



Gráfica 20: Rueda 1, Vel. 90, 1ª

Presupuesto



Gráfica 23: Rueda 1, Vel. 90, 2ª

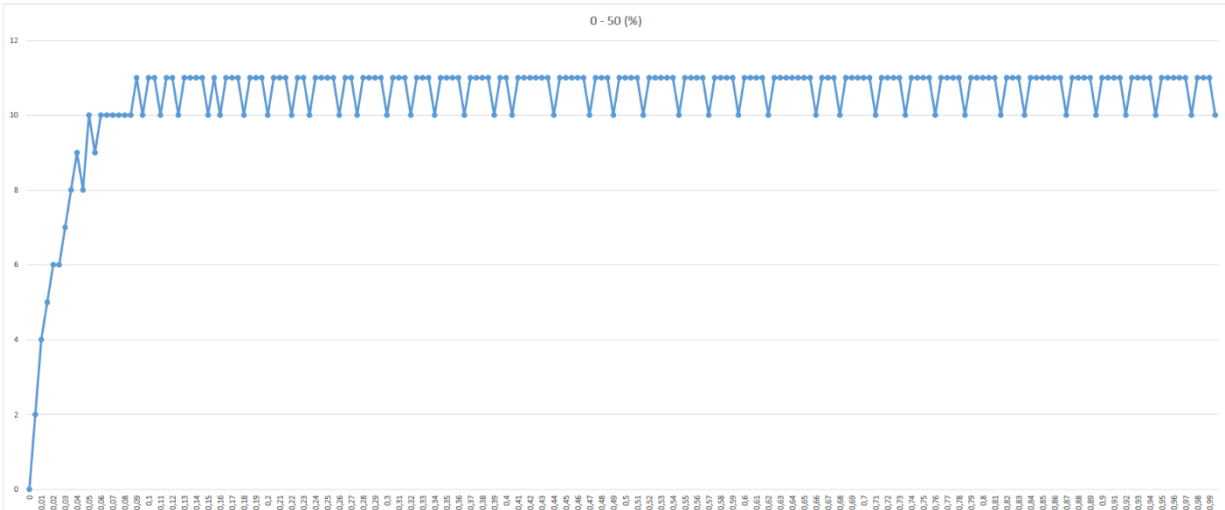


Gráfica 24: Rueda 1, Vel. 90, 3ª

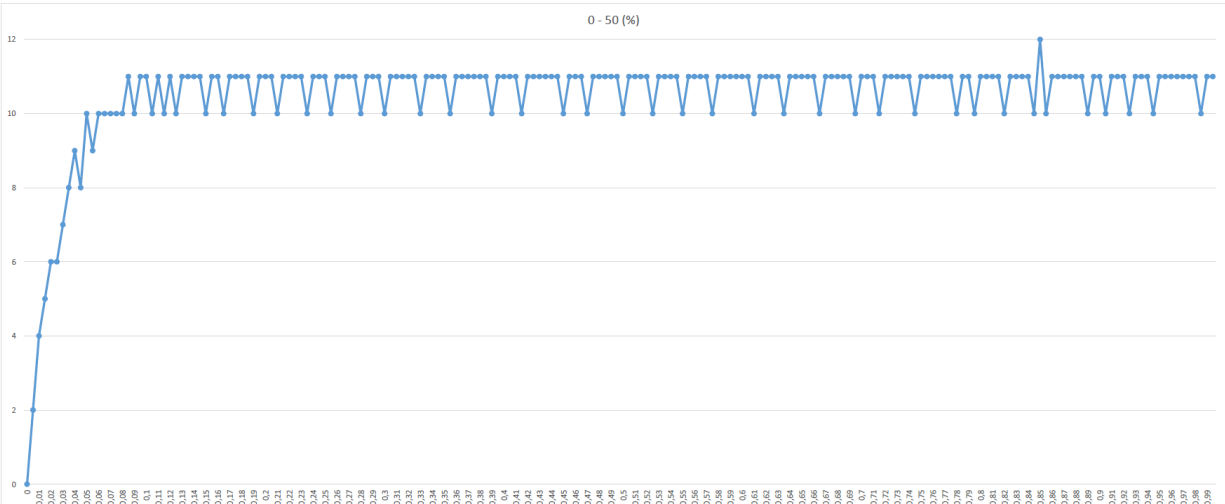


Gráfica 22: Rueda 1, Vel. 90, 4ª

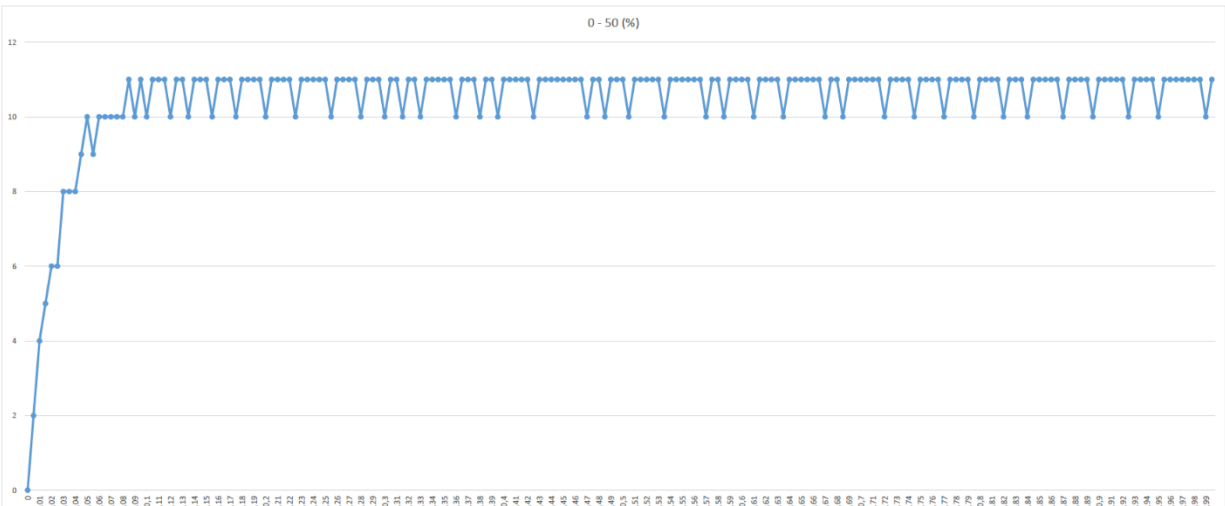
(2) Rueda 2



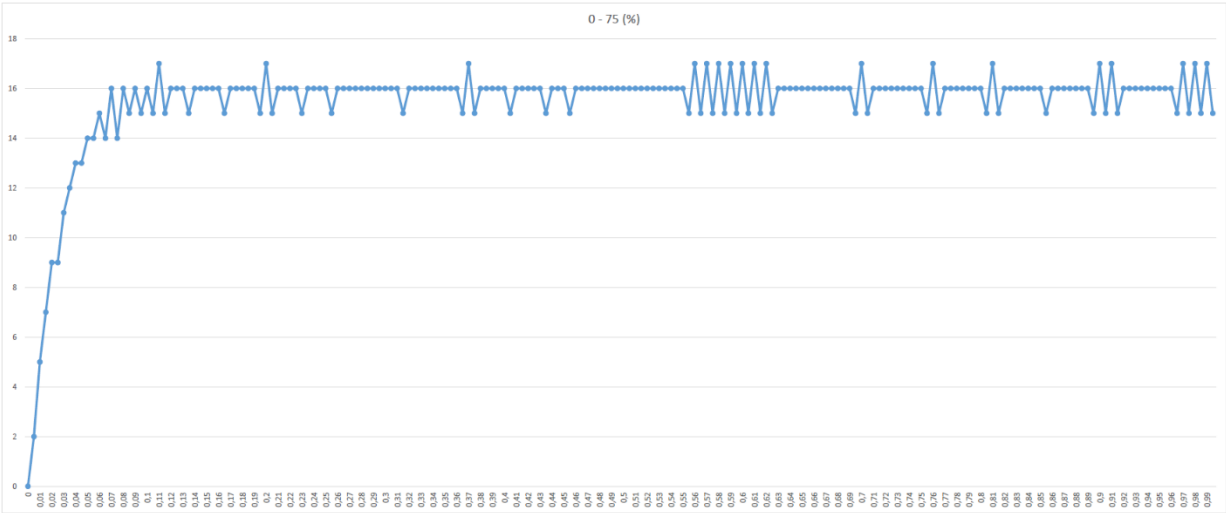
Gráfica 25: Rueda 2, Vel. 50, 11



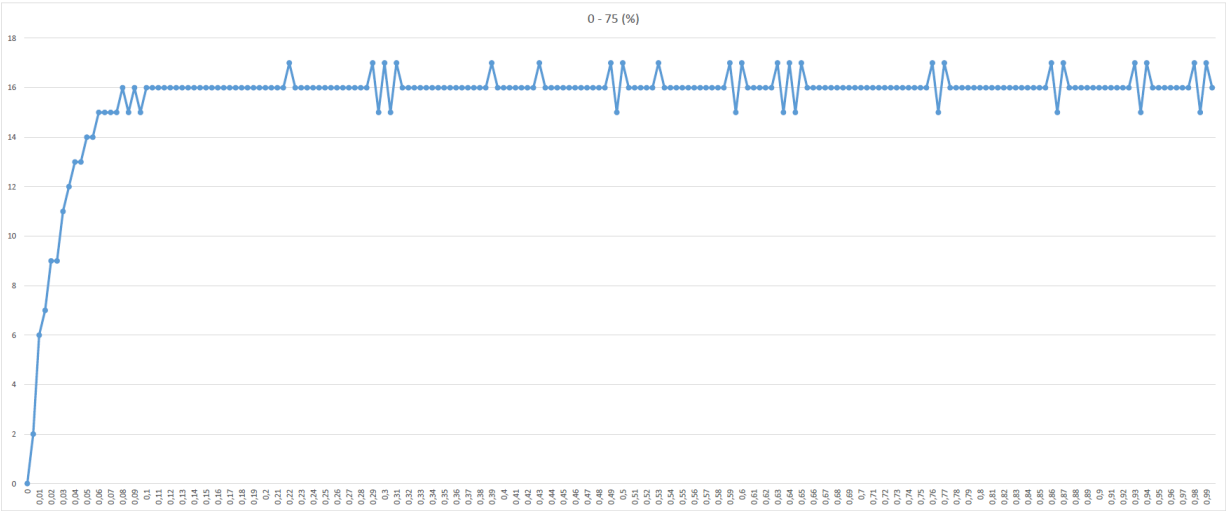
Gráfica 27: Rueda 2, Vel. 50, 2º



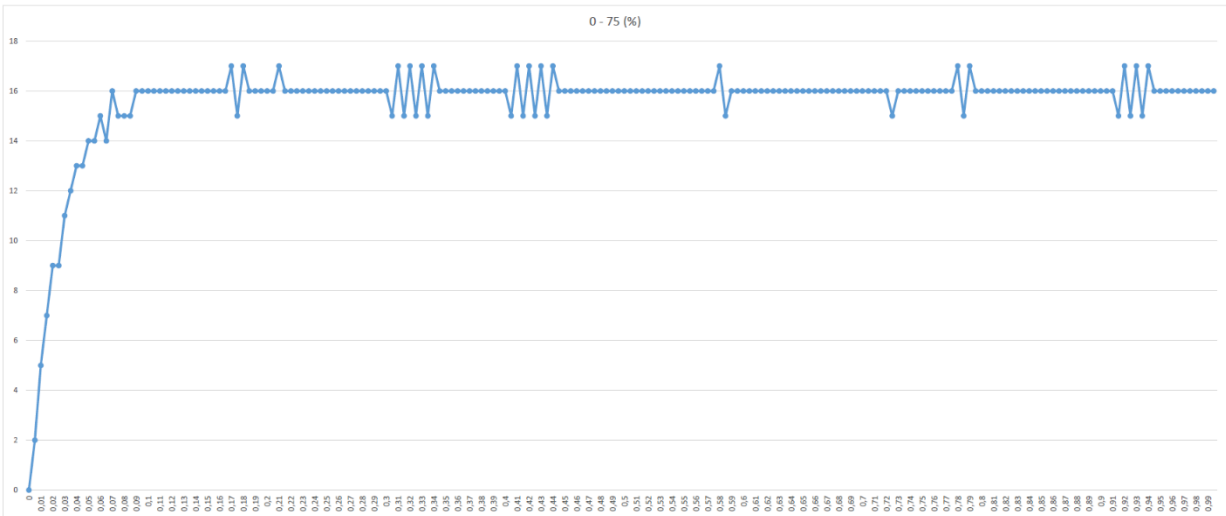
Gráfica 26: Rueda 2, Vel. 50, 3ª



Gráfica 29: Rueda 2, Vel. 50, 4ª



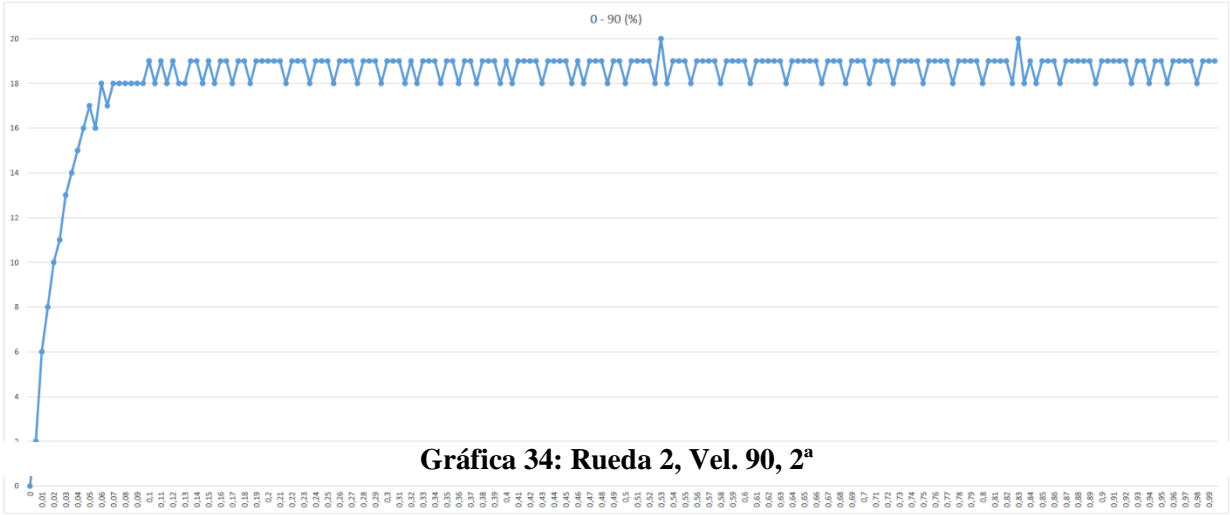
Gráfica 31: Rueda 2, Vel. 75, 3ª



Gráfica 32 : Rueda 2, Vel. 75, 4ª



Gráfica 34: Rueda 2, Vel. 90, 2ª



Gráfica 36: Rueda 2, Vel. 90, 3ª



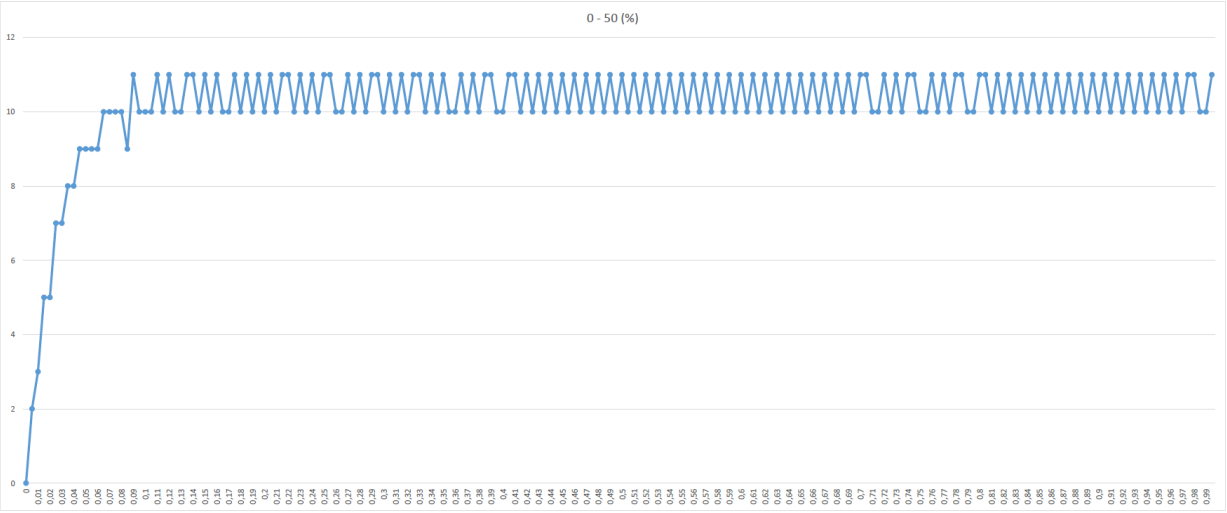
Gráfica 33: Rueda 2, Vel. 90, 1ª

Gráfica 35: Rueda 2, Vel. 90, 4ª

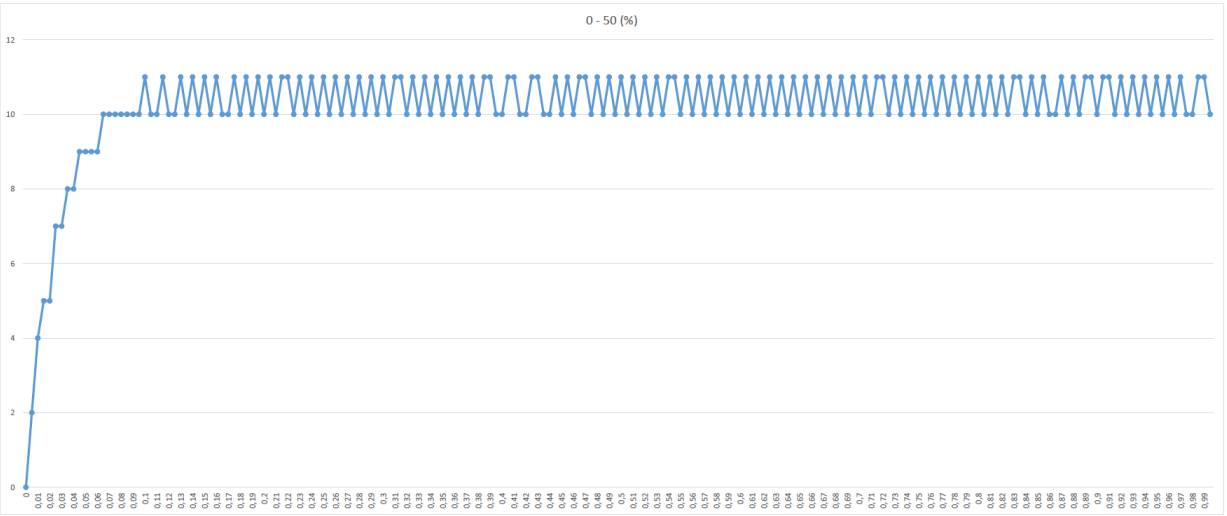
(3) Rueda 3



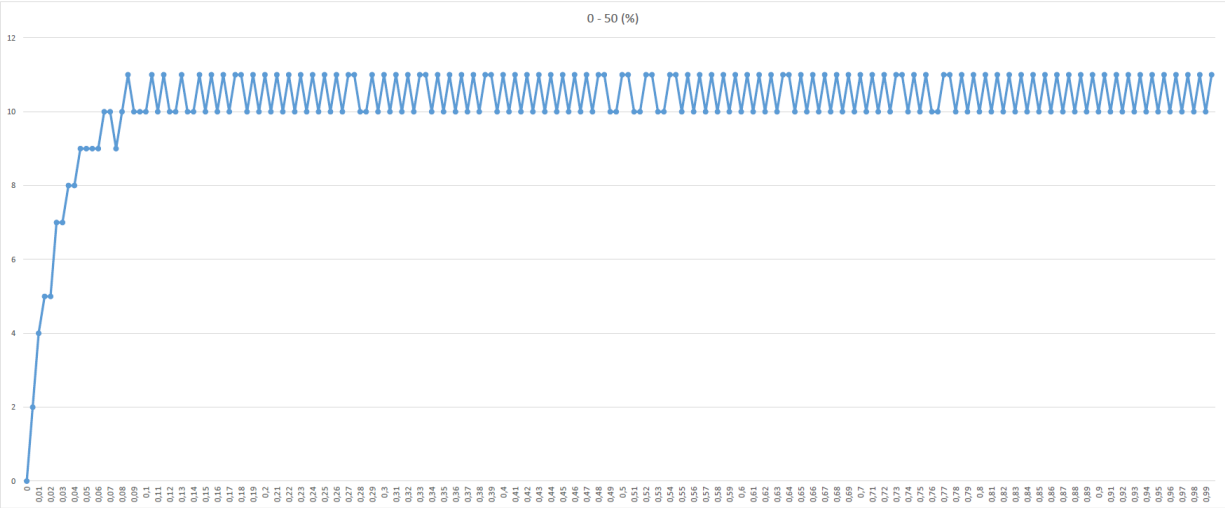
Gráfica 37: Rueda 3, Vel. 50, 1ª



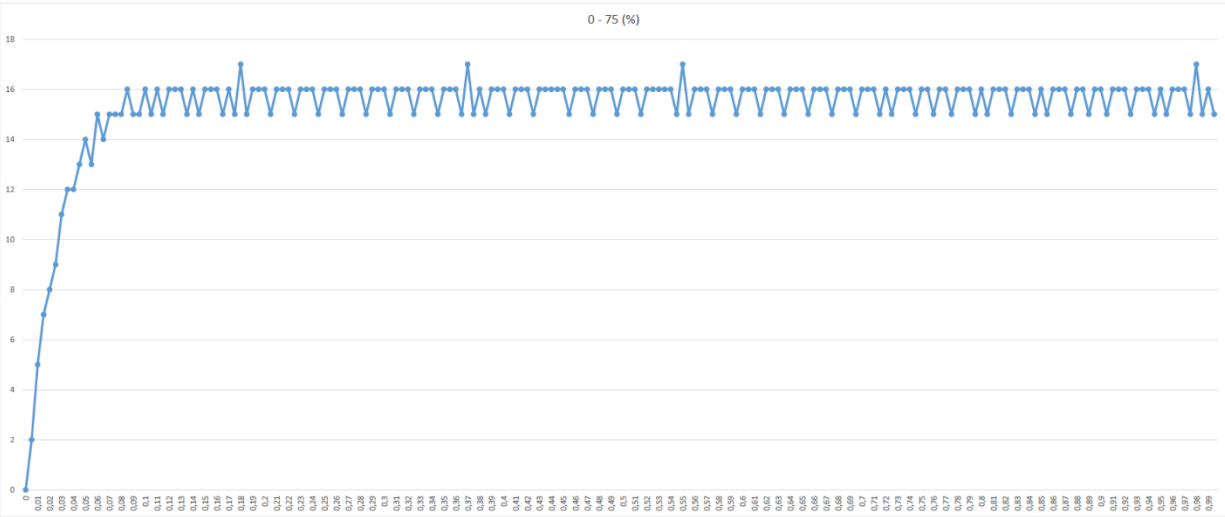
Gráfica 39: Rueda 3, Vel. 50, 2ª



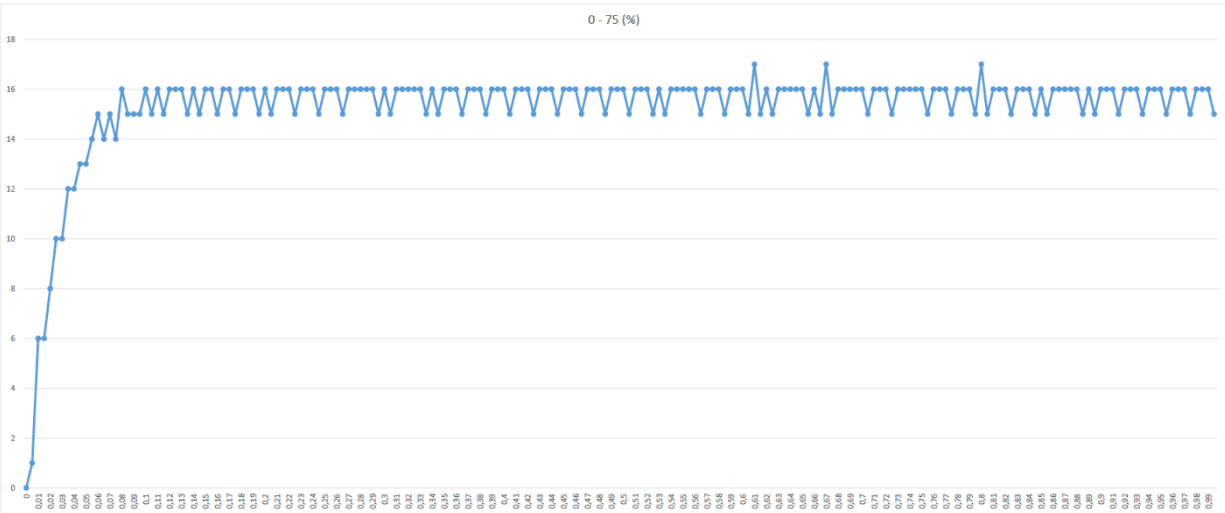
Gráfica 38: Rueda 3, Vel. 50, 3ª



Gráfica 41: Rueda 3, Vel. 50, 4ª

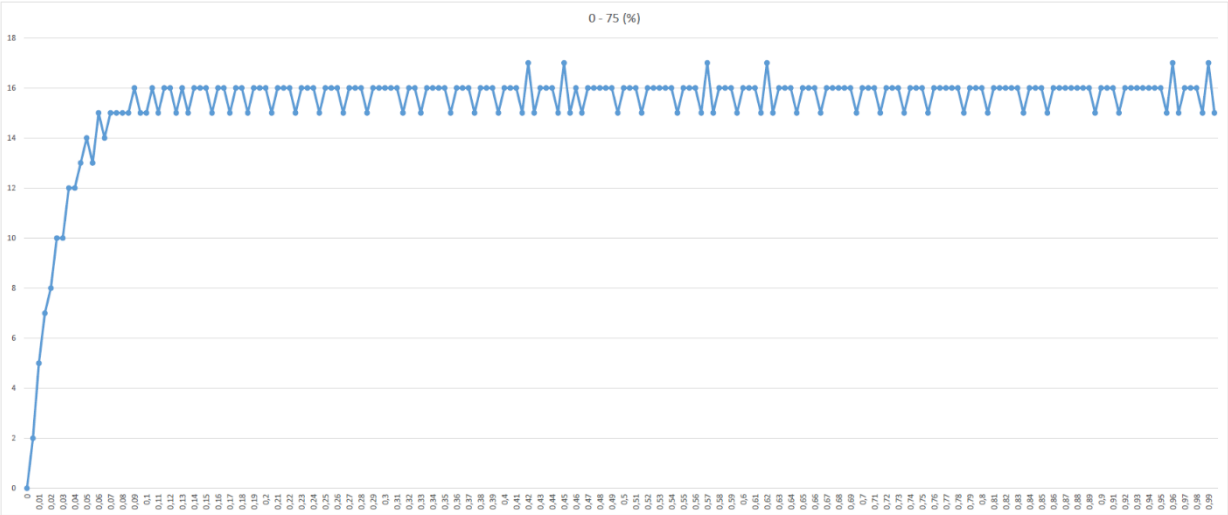


Gráfica 42: Rueda 3, Vel. 75, 1ª

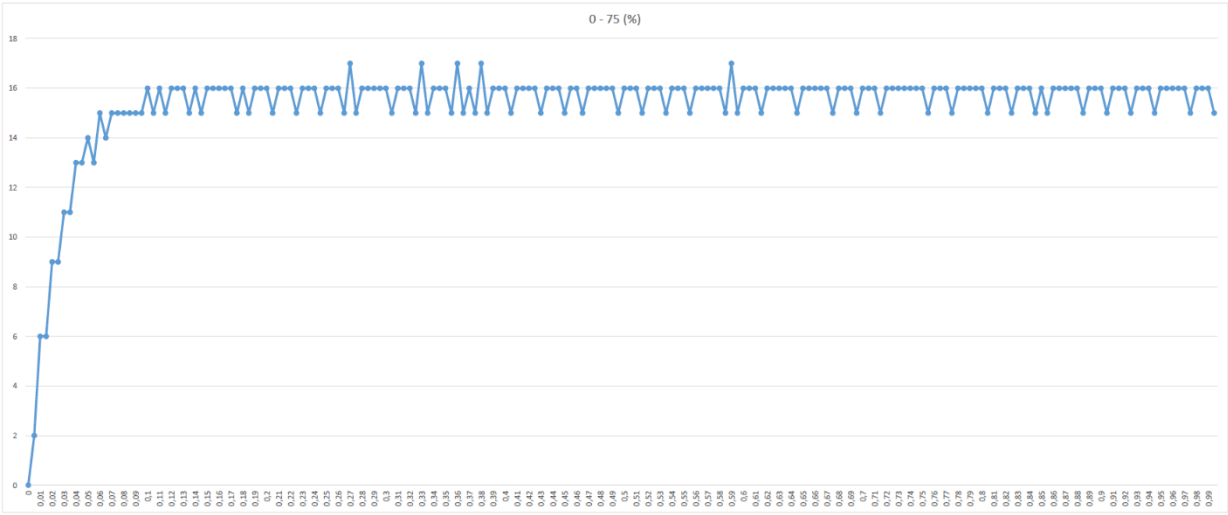


Gráfica 40: Rueda 3, Vel. 75, 2ª

Presupuesto



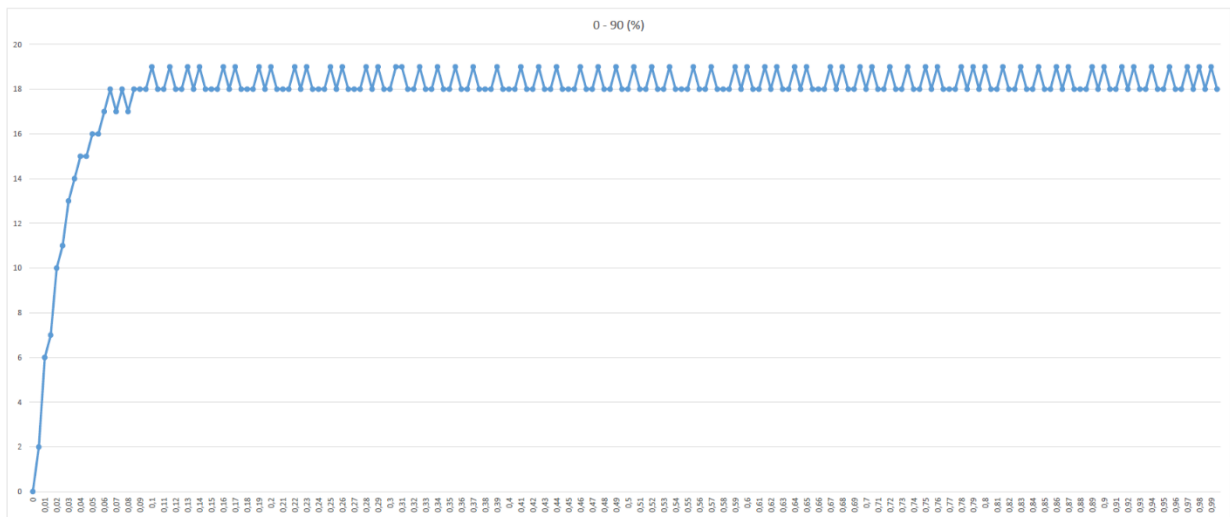
Gráfica 43: Rueda 3, Vel. 75, 3ª



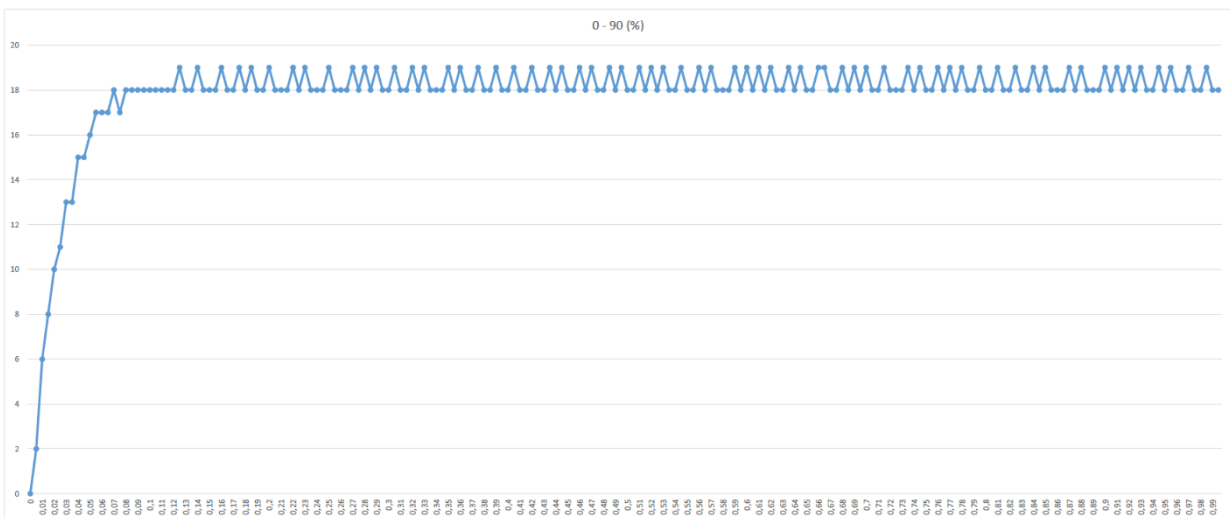
Gráfica 44: Rueda 3, Vel. 75, 2ª



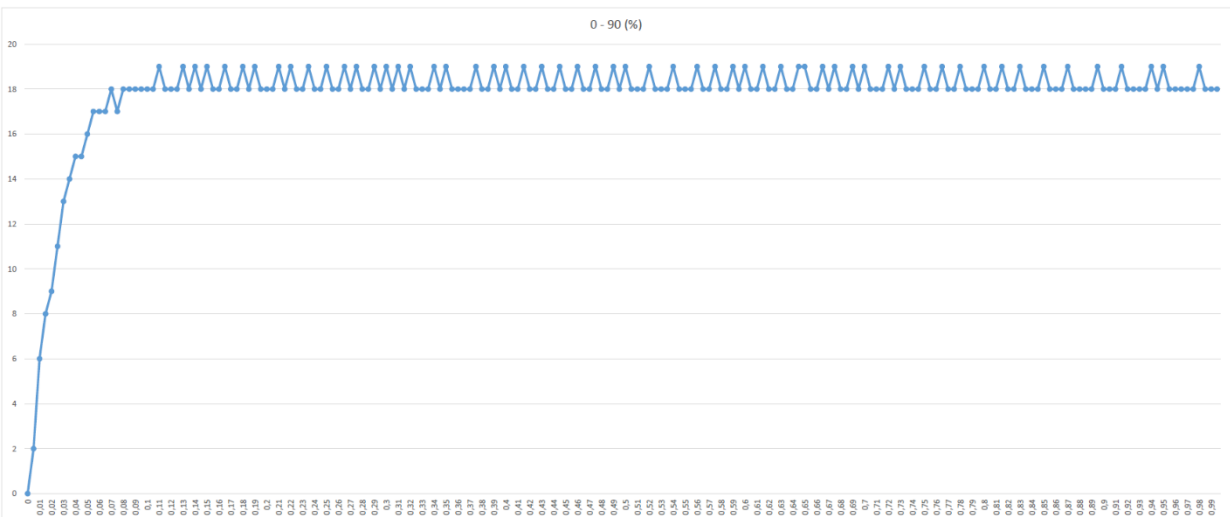
Gráfica 45: Rueda 3, Vel. 90, 1ª



Gráfica 46: Rueda 3, Vel. 75, 2ª



Gráfica 47: Rueda 3, Vel. 75, 3ª

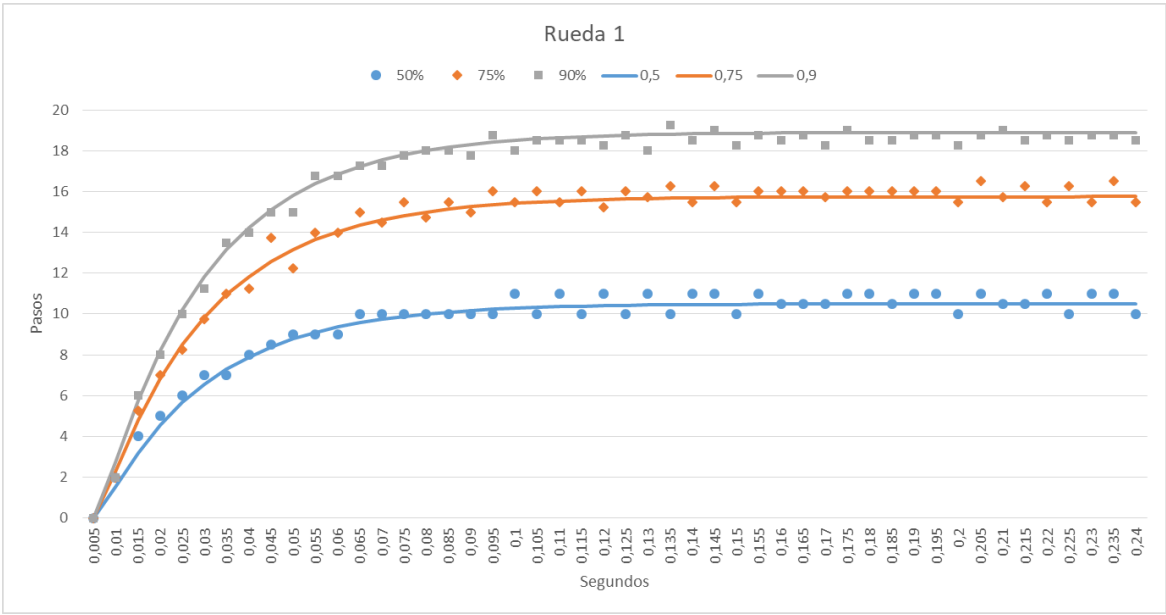


Gráfica 48: Rueda 3, Vel. 75, 4ª

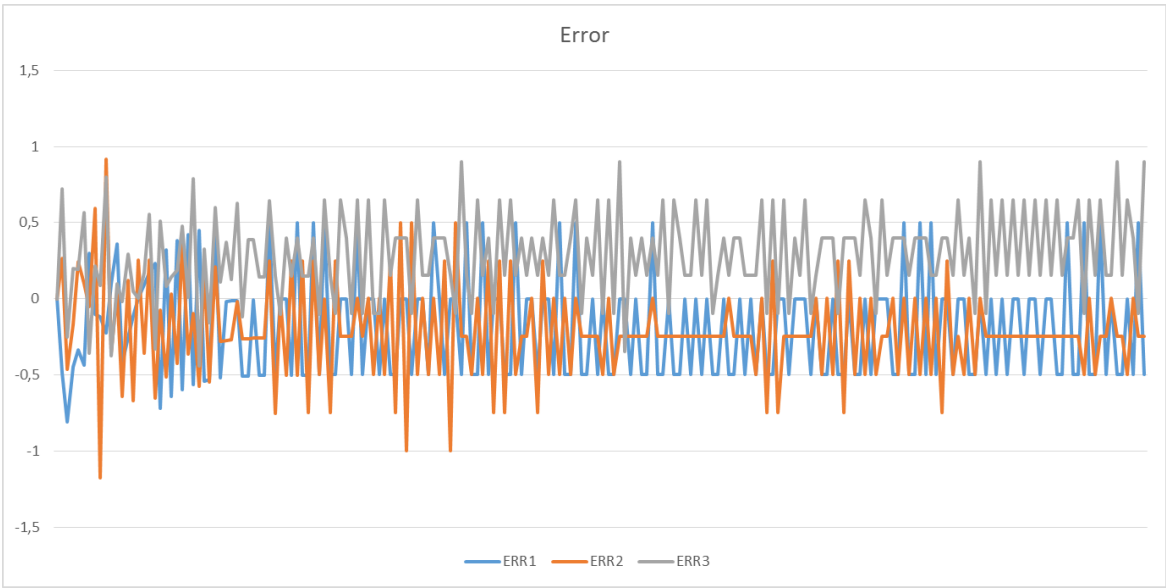
C Ajuste de las gráficas

Con el promedio de los datos tomados para cada porcentaje (50%, 75% y 90% de la velocidad máxima), se ajusta el modelo como se ve en las siguientes gráficas: Gráfica 49, Gráfica 51 y Gráfica 53. Con el consiguiente error: Gráfica 50, Gráfica 52 y Gráfica 54. La gráfica de la velocidad llega a 0,24 segundos, mientras que la del error llega a 1 segundo.

C.1 Rueda 1

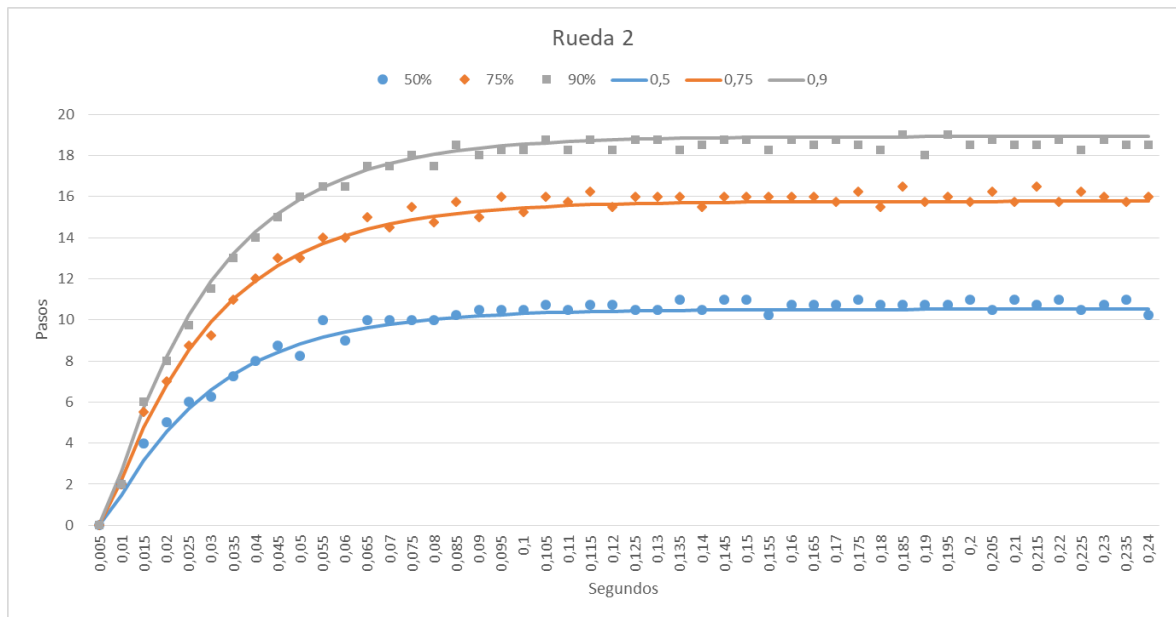


Gráfica 49: Ajuste de la gráfica de la función de la rueda 1



Gráfica 50: Error del ajuste de la función de la rueda 1

C.2 Rueda 2

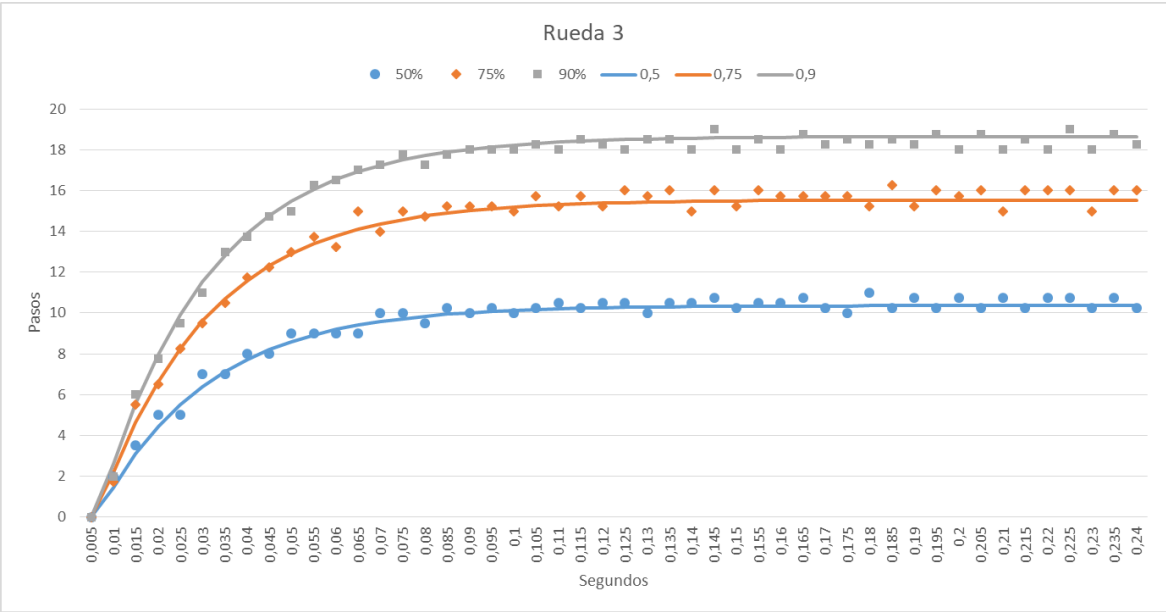


Gráfica 51: Ajuste de la gráfica de la función de la rueda 2

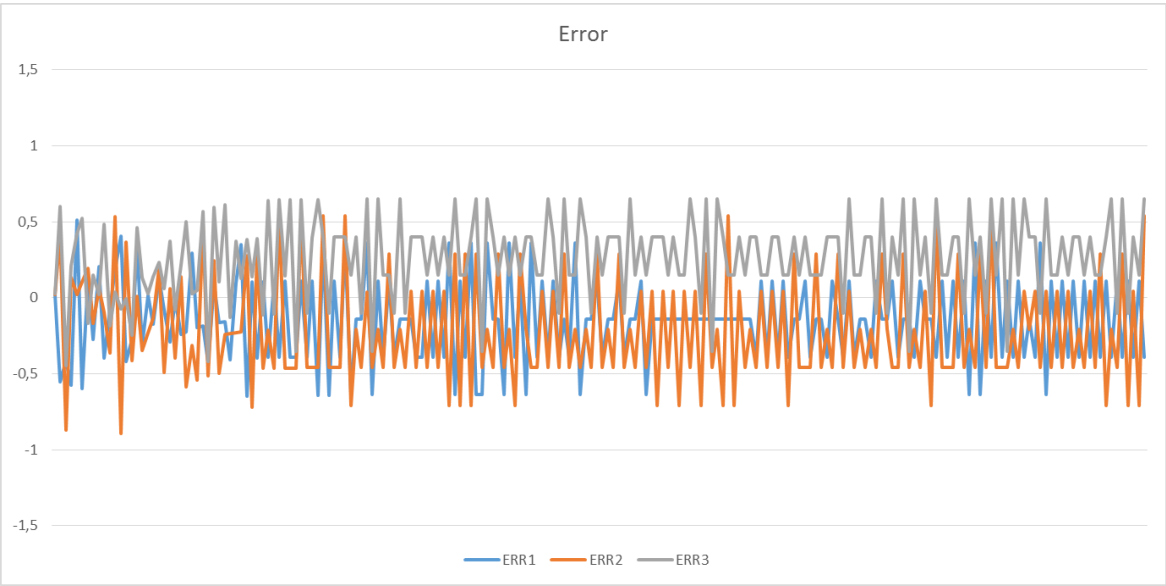


Gráfica 52: Error del ajuste de la función de la rueda 2

C.3 Rueda 3



Gráfica 53: Ajuste de la gráfica de la función de la rueda 3



Gráfica 54: Error del ajuste de la función de la rueda 3

